Computing today

FREE 12-PAGE SUPPLEMENT:
Adventure games — how to write them, how to solve them

Simple word processor for your BBC Model B

Using the ZX LPRINT for better quality printing from your Spectrum

Hardware review of the Laser 200, the latest Spectrum rival

A Stack Of Information On FORTH!

New programming series ● Software reviews ● Jupiter Ace dictionary

CONTENTS

VOL 5 NO 9 NOVEMBER 1983

Computing Today is constantly on the look-out for well written articles and programs. If you think that your efforts meet our standards, please feel free to submit your work to us for consideration.

All material should be typed. Any programs submitted must be listed (cassette tapes and discs will not be accepted) and should be accompanied by sufficient documentation to enable their implementation. Please enclose an SAE if you want your manuscript returned, all submissions will be acknowledged. Any published work will be paid for.

All work for consideration should be sent to the Editor at our Charing Cross Road address.

4

CONSUMER NEWS

A PEN FOR YOUR THOUGHTS ▼

A new company, Datapen Microtechnology Ltd, has been formed to manufacture and market computer peripheral equipment for personal computers. The first product to be released is a professional light-pen complete with a suite of programs.

Datapen's sales director Pam Rayner says "Most light-pens on the market are difficult to use because the operator has little or no control over the pen's data capture as the pen approaches the screen. Our light-pen has a LED readout showing that data is available and a switch built into the casing that allows the operator to signal when the position is right".

In addition to being absolutely insensitive to ambient lighting, the Datapen light-pen will respond to a wide range of screen intensities and colours.

Datapen is giving away a range of software with each light-pen. In addition to the expected user routines and example programs, Datapen is providing a 'Freehand Drawing Program' and is currently awaiting a technical drawing program that allows symbols to be created in a library and then combined on the screen.

Versions of the light-pen are currently available for the BBC, Dragon 32, and the VIC 20 micros, at £25.00 inclusive. Datapen Microtechnology is planning to introduce other products including two bar code readers, a digitising pad and a mouse device, all suitable for popular microcomputers and are looking for good dealers to support the products, both in the UK and overseas. Initially marketing will be by mail order from Overton. For further details contact Datapen Microtechnology Limited, Kingsclere Road, Overton, Hants RG25 3JB (telephone 0256 770488).

HANDY NEWS

Microwriter Ltd has announced a dramatic cut in the cost of their hand-held five-key personal wordprocessor — the Microwriter will now cost £299. This price slash of more than one-third brings it within the reach of most 'personal' purchasers as well as 'company' buyers. If you can manage to get into the International Business Show, there will be demonstrations of some of the text production applications for which Microwriters are proving valuable business aids. These include communication with a selection of the dozens of automated office machines with which it can interface, including microcomputers, electronic typewriters, printers, cassette recorders, electronic telex and wordprocessors. At the IBS, Microwriter will also be demonstrating the first alternative applications for the five-key keyboard, encouraging visitors to their stand to try their hand there and then. Graphic screen demonstrations, operated through standard Microwriter keyboards, will indicate Microwriter's first steps into new markets.

The Microwriter has successfully established its place as a truly portable personal wordprocessor since its national UK launch in July 1982. Many new Microwriter users are learning to Microwrite the letters of the alphabet, numerics, some punctuation and editing commands in just a couple of hours. At the Acorn User Show, even the editor managed to pick up the alphabet quite quickly, so it **must** be easy! With around 150 appointed Microwriter Centres throughout Britain, it is now reaching individual writers at High Street level nationwide. For further details please contact: Microwriter Ltd, 31 Southampton Row, London WC1 (telephone 01-831 6801).

ARISE, SIR ROM

It is more than likely that in the near future a great deal of ROM-based software will become available for the BBC Microcomputer. There are over 40 pieces of firmware already existing in ROM including languages, spreadsheets, financial and accounting packages, word processors, educational programs, games and file systems. Due to the provision of only four ROM sockets on the main BBC circuit board it is difficult to fully utilise the firmware now on the market.

Sir Computers can now supply an extension circuit board, fitting easily within the BBC case and capable of holding 12 extra ROM chips. When used in conjunction with the four existing sockets this provides 256K of memory (sideways-paged in blocks of 16K). This provides ample space for all but the most determined ROM collectors!

Fitting requires no soldering. The board plugs directly into an existing socket in the machine. The SIR ROM Board was launched at the Acorn User Show and is available for £35 (plus VAT). For further information contact SIR Computers Ltd, 91 Whitchurch Road, Cardiff (telephone 0222 21341/2).

CASSETTE CASE ▼

Computer users can now organize and protect up to 18 digital (or audio) cassettes with a smoke-tinted acrylic desktop file available through Inmac. The Cassette File is comprised of 18 separate cells, providing individual protection for tapes. The slanted design facilitates easy identification, access and removal of tapes, while the heavy gauge acrylic top protects tapes from dust and liquids. The file will also store Atari and TI game cartridges, and is under a one-year Inmac warranty. Non-slip feet provide extra security when the unit is used on smooth surfaces.

Further details can be found in Inmac's free, full colour catalogue along with details of nearly 1000 other items for computer users which are all available on a 30 day risk-free trial period and next day delivery. Please contact Inmac UK Ltd, Davy Road, Astmoor, Runcorn, Cheshire WA7 1PZ. (telephone Runcorn (09285) 67551: telex 629819 INMAC G).

QATAR SURPRISE ▲

Another first in the Arab world; Ramez M. Alhalaby, President of Autoram Computers headquartered in Jeddah, Saudi Arabia, has developed the answer for Arab Sinclair ZX81 home computer enthusiasts. Now, the ZX81 home computers can be programmed in Arabic. Prior to this, all Arab home computer users had to program using English words. Other computers with similar features are only available in large business computers, far too expensive for the average person. "There are many Arab people who have been waiting a long time for a full-Arabic computer. Now we have it," Alhalaby said.
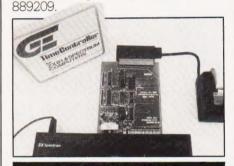
HAVE YOU GOT THE TIME ▶

Glanmire Electronics have just released the Time Controller for the ZX81 and Spectrum Computers. This piece of equipment consists of a battery backed-up real time clock with eight programmable inputs and eight programmable outputs. The clock provides the computer with the month, day, date, hours, minutes and seconds, and has its own built-in program, in PROM memory. Only a single instruction in the user's program is required to read or write the time and date. An extension for the RAMpack, printer and other peripherals is also included.

Applications include: electronic diary with alarm, home control, sophisticated burglar alarm, sound effects, electronic games, process control, and laboratory experiments. The recommended retail prices for the ZX81 and the Spectrum are only £34.50 and £38.50 sterling respectively: these products are presently available directly from

Glanmire Electronics Ltd: for further information contact them at Westley House, Trinity Avenue, Bush Hill Park, Enfield EN1 1PH (telephone 01-366 3245).

Enquiries from interested dealers or distributors are welcome. Generous quantity discounts are available. For further information please contact Glanmire Electronics Limited in Ireland: telephone Cork (021) 889209.

CHEAPER MONITORS

Data Efficiency Ltd, sole UK distributor for Kaga Monitors, is now offering the full range at significantly reduced prices and adding the extra benefit of 12 months warranty. The retail price of a Kaga 12" high resolution green monitor is now only £109.00 and is available with amber phosphor, while the 12" RGB medium resolution colour monitor is £239.00. The range also includes high resolution and super high resolution colour monitors at £285.00 and £399.00.

The monochrome monitors feature a non-glare tube as standard. The colour monitors connect to the Apple II or IIe by a separate interface card and to most other microcomputers including Apple III, IBM-PC and BBC by a single cable. For further information, please contact Data Efficiency Ltd, Maxted Road, Hemel Hempstead, Herts HP2 7LE (telephone 0442 60155).

BRIEFING

Astronics are offering an extended warranty scheme which, for a small annual fee, extends the maker's guarantee. The warranties are for three or five years, and are available not only for new equipment, but also for items on which the maker's guarantee has expired. The scheme covers many small micros including VIC 20, VIC 64, BBC, Dragon, Lynx, Oric etc, and can also cover peripheral equipment such as printers and disc drives. Warranties are available nationwide from micro dealers, or direct from **Astronics Ltd,** Unit 2, Eastern Industrial Estate, Jackson Close, Cosham, Portsmouth, Hants, PO6 1QW (telephone 0705 3266223).

A plain paper **listing service** for the ZX Spectrum is now available, which will print your BASIC program on white 80-column listing paper. It also includes all block graphics and underline characters. Any user-defined graphics are printed as the equivalent upper case letter, which is underlined: the UDGs will then be shown separately at the end of the listing, both as a printout of the actual graphic character, and a printout of the eight decimal bytes required to form the character. The address to send your program, instructions and remittance to is **TPA Computer Services,** 189 Upland Road, London SE22 0DG. Prices are as follows: £1.50 per program (75p for duplicates); minimum order value £3.00. The price includes postage and packing, and all cassettes and Microdrive cartridges are returned.

Priced at around £350, with full CP/M implementation, 64K of ROM and 32K of RAM the new **Triumph Adler Alphatronic PC** will be attractive to business and professional users as well as the amateur who has out-grown his hobbyist micro and now wants to upgrade to a more powerful system. The PC's CP/M implementation gives compatibility with each of the Alphatronic range of business micros. The user may opt for floppy discs (one or two, each with 320K storage) or cassette storage. A wide range of interfaces come as standard, including RS232, Centronics and Kansas City I/O. A wide range of applications available on the PC includes software for games, accounting, wordprocessing, education, training and scientific applications. The Triumph Adler Alphatronic PC will be available from Triumph Adler dealers throughout the UK.

NEXT MONTH

Computing today

COMMODORE'S 64

Anyone who has purchased a Commodore 64 will be aware of two things — the machine has been designed to provide some impressive graphics and sound capabilities, and the BASIC has been designed to make it as difficult as possible to actually use them. Well, that's how it appears, when everything has to be controlled by POKEing and PEEKing all manner of memory locations in various parts of the machine. Next month *Computing Today* starts a three-part series that looks at the sound and graphics functions, explains exactly what's going on and where, and helps to the programmer use features to the full. We start next month with a look at sprites, and there are two useful machine code routines to allocate memory space for sprite data and move sprites about the screen. You can't afford to miss this one.

MIKRO MAGIC

Of course, with everything in the 64 controlled in this manner, some programmers will find it a lot simpler to dispense with BASIC altogether and write directly in machine code. This calls for a well-written editor, assembler and monitor (unless you're a masochist who codes in hex on bits of paper), and the December CT has a review of the MIKRO Assembler cartridge from Supersoft. At £50 it's not cheap, but does it provide value for money?

DRAGON DRAWING

Getting to grips with the high-resolution mode on the Dragon 32 is often so tedious (all those points to calculate and plot for a complex shape!) that it's a rather neglected feature of the machine. Our 'Graphicks' program is a versatile piece of software that allows freehand drawing on the screen using a joystick, the placing of circles, rectangles, lines and blocks anywhere on screen and in any size, turtle graphics, and the ability to load and save screens or dump them to a printer. A must for frustrated artists.

Articles described here are in an advanced state of preparation but circumstances may dictate changes to the final contents.

MC-BASIC CONVERTER

One of the more common methods of mixing BASIC programs with machine code routines is to store the machine code instructions as DATA statements and include a subroutine to POKE the data into memory as required. This method is simple but you have to convert all the hex codes to decimal (plenty of scope for errors), and then type in the DATA statements (more errors). For Genie owners, at least, this is no longer the case — with the program we'll be publishing in the next issue, you can develop the machine code and get the computer to convert it automatically and store it in the BASIC program lines as DATA, quickly and error-free. Pretty neat.

COMPETITION NUMBER 1

Next month we'll be running a competition with a rather unusual prize. We're not going to give away the secret just yet: walls have ears, and in any case you would'nt want us to take all the suspense out of life, would you? Let's just say that fame and fortune could be yours, and it might be a good idea to read the supplement this month rather carefully.

?

GIVE THIS TO YOUR NEWSAGENT
Please reserve me a regular copy of
Computing Today

NAME ....
ADDRESS ....
POSTCODE ....

SOFT WARES

ALL AT SEA ▲

The Epson HX-20 portable computer has been used as a basis for a coastal navigation program written by Devon yachtsman Tony Whittaker. According to him, the HX-20, which was supplied by Southern Computers of Torquay, is something for which he has been waiting a long time, and is the only computer that has anything like the facilities he needs.

Since completing the 18K + program, which requires the additional 16K RAM pack to be used, it has been on trips from Salcombe to the Helford River and back by way of St. Mawes and Fowey and from Salcombe to Lezardrieux, St, Malo, Iles Chausey and St. Peter Port. "The HX-20 has come through its sea trials with flying colours, even surviving unscathed an aerial journey from the chart table to the cabin when a particularly nasty wave off Bolt Head laid my yawl over on its side" said Tony Whittaker.

At present he is discussing the possibility of marketing the program worldwide with international yachting people, one of whom is testing the program at the moment. Details of the program can be obtained from A.M. Whittaker, The Court House, Moretonhampstead, Devon. (Anyone want to bet that the Americans will decide it's illegal?)

EDITING ON AN EPSON

The portable Epson HX-20 microcomputer can now be used as a sophisticated word processor, following the introduction this month by Ffoss Ltd of its Ffosswriter word processing package. Ffosswriter is designed to work with either the standard or expanded HX-20 as well as with the internal printer or an external dot-matrix or letter-quality printer. Supplied with the program, which is contained on an MC30 cassette, is a plug-in EPROM that gives the microcassette tape drive a random access facility similar to that of a disc. The user can operate simply with named documents and forget about how the tape is organised.

In brief, the main facilities of Ffosswriter are:
Document handling (eg directory, create, edit, resume, delete, save, quit);
Editing (using conventional insert, delete and cursoring features);
Word Processing functions (eg part document editing without reading whole document; define block by start and end for text expansion; delete or print a block, and move or add blocks to any document from any document);
Page Layout (eg width, length, margins, tabs, indents, page numbers, line spacing, bold, condensed, enlarge, super/subscripts, underline, right ragged or justified, and line centreing.)

Ffosswriter allows line lengths of up to 80 characters to be displayed on the HX-20's built in 20-character by four-line screen. Whole lines are shown, not by a window method but by wordwrapping over the screen's 80 character display. (Display adaptor software will be available from Ffoss sometime in the future to permit Ffosswriter to be used with a separate larger screen for desktop operation.) The software comes in four basic modules: EDIT (edit and internal print), PRINT & TRANSMIT (external print and transmit), INSTALL (install an external printer), and FORMAT (formatter for data tapes). The modules are arranged in two 14K blocks to allow users with the standard HX-20 to load the required modules one at a time

into the machine's user RAM. Machines with Epson's 32K RAM expansion unit can be loaded with the complete program in one operation.
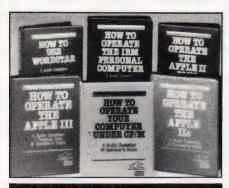
Supplied with the Ffosswriter software is a 50-page A5 spiral bound user manual, a quick reference card of the more common commands, an overlay for the user function keys — all enclosed in a video cassette style library case which has seven spare slots for the user's document tapes. The cost of the total package is £95 plus VAT (inclusive of post and packing for mail orders.) Generous quantity discounts are available for trade orders. As a special offer to customers producing a HX-20 User Club membership card, Ffoss will deduct £10 from the recommended retail price. For further information please contact Ffoss Ltd, 112 Bath Road, Slough, Berks SL1 3SZ (telephone 0753 820277).

DONGLE DEVELOPMENTS ▼

Dongles, the software protection devices developed by Wordcraft Systems for use with its own word processing software, are now available for use with IBM-PC, ACT Sirius 1 and Commodore machines. Each dongle contains a unique coding which is interrogated by the software during program operation. If the dongle is not present or its coding differs from the serial number embedded in the software, the program will not function. Users can make security copies of their discs, but the need for the dongle to be present at run time confines their use to a single machine.

Wordcraft Systems has always made the dongle available to other software houses, and guarantees that the devices are unique to each company or product. Dongles for the IBM-PC fit onto the parallel printer port and allow for in-line connection and normal operation of the printer. The cost of the dongles ranges from £2.50 for Commodore VIC 20 machines to £15.00 for IBM-PC versions; minimum order quantity is 100. For further information contact Wordcraft Systems, 43 Farley Road, Derby DE3 6BW (phone 0332 683892).

PLAY IT BY EAR ▲

For the busy professional who is new to computing and finds operating manuals less simple to use than they claim to be, FlipTrack Learning Systems, just introduced to the UK by Marketing Micro Software Ltd, could be the answer. Designed for the first time user of a CP/M operating system, the IBM PC, Apple II, Apple III, Visicalc or Wordstar/Mailmerge, these systems are available as packs of three or four audio cassettes which take the mystery out of the manual approach and allow the beginner to learn by doing rather than reading.

Colin Aldridge, MD of MMS, points out, "Many new computer users are understandably impatient to master their systems or to get the most out of an investment in a new software package. Many manuals are off-putting. It is difficult to follow a manual and try out what you are learning on the keyboard simultaneously. Audio systems speed up the learning process by allowing the beginner to try out the various stages to mastering the system concerned as he or she goes along."

FlipTrack systems are available from leading computer dealers or direct from MMS, Goddard Road, Whitehouse Industrial Estate, Ipswich, Suffolk IP1 5NP (telephone 0473 462721).

ADDRESSBOOK ADDENDUM

For those of you who read our sister magazine *Personal Software*, the Summer issue contained an article for the Commodore 64, entitled Addressbook.

We would like to point out here that this article has no connection whatever with a mailing list management package called Addressbook which is supplied by Decision Technology. This commercial package is available under CP/M and for the IBM PC.

We apologise for any confusion that may have arisen over the title. If you want more details on the commercial Addressbook, contact Decision Technology, 7 St Johns Road, East Molesey, Surrey KT8 9JH.

GETTING THE HOBBIT HABIT

The Hobbit, the top selling program for the Sinclair ZX Spectrum, is soon to be available for the BBC, Oric and Commodore 64 computers. Managing director of Melbourne house Alfred Milgrom explained that, due to the success of the program, The Hobbit was being adapted to suit other popular microcomputers. Melbourne House Hobbit, which includes the program, an instruction booklet and a copy of the classic J. R. R. Tolkien novel on which the adventure is based, will go on sale in mid-September, in time for pre-Christmas sales. Further adaptations for other micros will follow soon after.

The Hobbit is available from all leading retailers or by mail order from Melbourne House, 131 Trafalgar Road, Greenwich, London SE10. Recommended retail price is £14.95.

FOR THE LOVE OF ADA

Two systems and software houses have announced the availability of a new compiler for the Ada programming language. York/Ada, a VAX-based compiler, is the result of a four-year development programme by the University of York, Europe's leading academic institution in Ada. This announcement is seen as a major step forward in ensuring the widescale use of the programming language in defence, technical and commerical applications. SPL International and Software Sciences Ltd, both founder members of the Ada Group Ltd, will jointly market the compiler worldwide. VAX UNIX and VAX VMS versions of the product are available immediately and other implementations will follow.

The compiler has already been in use in 12 establishments for more than a year. For further information please contact Software Sciences Ltd, Abbey House, 282/292 Farnborough Road, Farnborough, Hants (phone 0252 544321) or SCMS, 347 City Road, London EC1 (phone 01-278 0950).

FREE FORTH THING

The FORTH Handy Reference Card is now available free from the FORTH Interest Group (FIG). Functioning as a pocket programming aid, the card lists and describes the major commands of the FORTH computer language. This card serves as an excellent reference for all FORTH programmers, beginner to expert. Commands are grouped by function for easy reference and use. These include stack manipulations, number bases, arithmetic and logical comparison, memory, disk handling, defining words, vocabularies, and system.

The FORTH Interest Group is a non-profit organization of over 40 chapters and 3,800 members worldwide devoted to the dissemination of information about the FORTH computer language. For further information and free Reference Card call the FIG HOT LINE at (415) 962-8653 or write to the FORTH Interest Group, PO Box 1105, San Carlos, CA 94070, USA.

BRIEFING

Audiogenic have produced a new catalogue, listing all their current programs for the VIC 20, CBM20, Spectrum and Dragon. The catalogue lists products and gives a brief and well written description of each. Prices are given, all of which include VAT. Copies of the catalogue are available direct from Audiogenic or via the nationwide dealer network. For further information, please contact **Audiogenic Ltd**, PO Box 88, Reading, Berks.

Audio Visual Supplies are marketing by mail order the recently released **Master Class** video training cassettes of 'How to use your Computer.' This series is intended to give a thorough grounding for users and teachers alike in how to use the BBC Micro. The video tapes bring your own personal tutor, David Redclift, who has been involved in the training of programmers for the past 12 years, into your home or school. He explains and demonstrates in an easy-to-understand way, how to write your own programs and use your computer to its full potential. Each video tape runs for approximately one hour and there are three tapes so far in the series: Starting to Learn BASIC with the BBC Micro: Further BASIC with the BBC Micro: The BBC Micro in Primary Education. The video cassettes cost £19.95 each (including VAT) plus £1.00 postage and packing and are available from **Audio Visual Supplies**, 71 Kemps Green Road, Coventry CV7 7QF (telephone 0676 33965).

BUSINESS NEWS

TINKERING WITH A GiPSI ▲

A unique new science education product is announced by Griffin & George, the education supply arm of Fisons Scientific Equipment Division. Considered by the company as their best-ever new product development, the new device combines, in a single instrument, the functions of at least 30 (initially) other instruments currently used in science laboratories. It also has wide potential applications in industry.

Called the GiPSI (Griffin Programmable Scientific Instrument), the new multi-purpose instrument will be in constant daily use, unlike existing single-function instruments that tend to lie idle on the shelf for long periods. Since it costs a fraction of their price, it will be of interest to all science teachers, who seldom have a big enough budget for much of the equipment they need. Another attraction of the new device is that it requires no special knowledge or skills to use.

An innovative application of microchip technology, the GiPSI is converted into the instrument you want in less than four seconds. It can be used as any one of at least 30 different instruments, such as a pH meter, scaler/ratemeter, oxygen meter, joule and watt meter, digital multimeter, timer, counter, frequency meter, data monitor, conductivity meter magnetic flux sensor, temperature sensor, angular and linear motion meter, pressure sensor, time-speed acceleration meter, and so on. In fact it will replace all the instruments on the shelves behind the young lady, using the bits and pieces on the desk in front. All you do is plug in the appropriate program module (seen at the far side of the table) and place the corresponding printed overlay (at the front left side of the table) over the the GiPSI keyboard and you're in business! An easy-to-read instruction manual guides the student through the simple steps required. Product enquiries should be made to the Technical Marketing Department, Griffen & George, Frederick Street, Birmingham B1 3HT (telephone 021-236 2552).

THE PLOT THICKENS ▶

Environmental Equipments (Northern) Limited have launched the latest digital X-Y plotter in their Digiplot range — the Digiplot II. The new plotter is an A3-size, six pen version and at just under £800, is claimed by Environmental Equipments (Northern) to be the lowest prices of its type available in the UK.

With a pen speed of 150 mm per second, the new Digiplot II is three times faster than the previous Digiplot single-pen and six-pen models it replaces. The speed can also be varied in 10 steps down from the 150 mm maximum to 15 mm per second.

The Digiplot II also has 28 intelligent functions — more than the previous models, including extra symbols such as circles, stars, diamonds, and so on. The Digiplot II also has the ability to specify overall length and number of divisions or the distance on the axis, which it then works out. Solid or broken lines can be specified with no limitation on the pitch of broken lines. Open or closed boxes can be hatched and the pens can be maintained within their correct path, even after being sent out of the plotting area. 254 different character sizes can be selected on the Digiplot II, the smallest being 0.7 × 0.4 mm, the largest 177.8 × 101 mm — a vast increase on the 16 sizes available on the earlier single pen and six pen models.

Another benefit provided by the new plotter is automatic raising of the pen after 4 seconds, if no other command has been programmed. This prevents pens from 'bleeding' onto the plotting surface, while in a stationary mode. For connection to a computer, the new Digiplot II offers you the ability to specify the appropriate interface — with a choice of RS232C, Centronics or IEEE interfaces available.

The new Digiplot II forms part of a comprehensive range of digital X-Y plotters — over 20 models varying in size from A4 to A0, with up to 10 pens and 45 intelligent functions. More details on the new Digiplot II or other plotters in the range can be obtained from Environmental Equipments (Northern) Limited, Environ House, Welshmen's Lane, off Welsh Row, Nantwich, Cheshire CS5 5ES (telephone 0270 625115).

WHAT A SIGHT

Digithurst Ltd have announced the release of an image analysis package to run on the Sirius microcomputer. The package is an enhanced version of Digithurst's successful MicroSight I Vision System. The enhancements include an increase in resolution to 256 by 256 pixels and the use of recursive filtering to reduce the effects of random interlace experienced with video cameras.

The MicroScale image analysis package has also been implemented on the Sirius and allows the user to manipulate images by moving and expanding sections of the screen as well as the usual features such as dimensioning and area or perimeter measurement. Applications include input to CAD, graphic design, image analysis, forensic analysis, low cost medical imaging and cartography. Costs are £495 plus VAT for the MicroSight, with the MicroScale packages costing from £295 plus VAT. More details can be obtained from Digithurst Ltd, Leaden Hill, Orwell, Royston, Herts SG8 5QH (telephone 0223 208926).

RML DISCS ▼

Research Machines has announced the availability of a new product — high quality disc drives for its top-selling schools systems the LINK 480Z. The introduction disc drives is another important step forward for the 480Z, which was originally designed to act as either a station on RML's CHAIN network or as a stand-alone cassette-based system. Characteristically, RML have done the job properly by allowing users to read and write in both single and double-density mode to both single and double-sided discs.

The units are available with either single or dual drives, and also contain an 'intelligent' disc controller plus power supply. The disc controller contains its own processor which takes care of the different modes of operation. In double-density mode, a single drive system has a capacity of 328 Kbytes of storage and a dual drive system offers 656 Kbytes. The system uses the standard CP/M 2.2 operating system and this provides complete compatibility with Research Machines 380Z systems. Thus the wealth of disc-based software that is available for the 380Z will be available for the LINK 480Z.

The single drive unit will retail from its launch in September at around £500 TEP (Typical Education Price). Primary schools which have bought a LINK 480Z under the Department of Industry 'Micros in Primary Schools' scheme will be given the chance to buy a single or dual drive unit at a special concessionary price, to encourage them to make the step forward to disc systems. For more information and the full pricing structure, please contact Research Machines Limited, Mill Street, Oxford OX2 0BW (phone 0865 249866).

COMPUTER POST ▶

A new national electronic mail service, which will deliver letters instantly for less than the cost of a second class stamp, has been launched by ACT. It is the first such national service for users of low-cost microcomputers — and it is named Micromail (how original).

Micromail is based on British Telecom's electronic mail service, Telecom Gold, which already supports several thousand electronic mail boxes. Subscribers will be able to transmit letters at the speed of light (a bit faster than a letter), with simultaneous copies to up to 500 people at no extra charge, and the system will link into international services so documents can be transmitted to micro users worldwide.

Micromail is cheap and efficient — a one-page A4 letter of 400 words, for instance, can be transmitted to its destination in less than a minute, at a peak rate price of 17p long distance and 15p within London. During off-peak times, the cost would be less than that of a second class postage stamp. The same letter sent by telex would cost three times as much and take six times as long to transmit. Facsimile transmission would cost five times as much; Datapost almost 20 times as much.

The Micromail package consists of a small modem card which plugs inside the computer; user-friendly software developed by ACT to enable it to communicate over ordinary telephone lines; and the annual subscription. Yet the package price is only £275 plus VAT. ACT has initially made Micromail available for users of its Sirius 1 and new Apricot microcomputers, and plans to support the IBM in the next few

months.

For further information contact Applied Computer Techniques Ltd, ACT House, Telephone Avenue, Bristol BS1 4BS (telephone 0272 211733).

BRIEFING

Commodore Business Machines (UK) Ltd have announced that the suggested retail selling price of the majority of their business micro computer range, disc drives and printers, will be reduced, on average, by over 25%. Typically, this reduces the cost of a Commodore micro computer business system (incorporating a 32K computer, a double sided, 2 Megabyte dual disc drive, a dot matrix printer and cables) from £2750 to £1985, excluding VAT — a reduction of 28%. Full details of the hardware price adjustments ae available from Commodore authorised dealers.

Zenith Data Systems has introduced a 256K dynamic RAM card, known as **Z205**, for use with its Z100 series of desk top microcomuters and other S-100 computers. The new memory card is a low cost means of increasing the power of Zenith's Z100 system by providing greater capacity for larger programs. The Z100 series provides 128K RAM as standard and can be expanded to a total of 768K user RAM. By adding the Z205 memory cards to the standard system, comprehensive business programs can be handled, such as accounting and stock control, without segmenting the programs. The user will no longer need to change a number of discs, greatly increasing the speed of operation. The cards can be simply plugged into the S-100 board within the computer on the users' premises. The Z205 256K dynamic RAM card is available from appointed Zenith dealers throughout the UK for £520 plus VAT.

Janet Ansell

BBC WORD PROCESSOR

Word processing packages are available for the BBC Micro in ROM, but it's possible (and cheaper) to use this program and get good results.

Having purchased an Epson MX-80F/T III printer for my BBC Micro I decided to write a Word Processor program that would make full use of it. Before writing the program I decided upon the following requirements. The first was that a user may enter or edit stored text. The minimum editing facilities must enable the user to add and delete text. Additional editing functions may also be provided such as text moving and search and replace.

The second requirement was that the text can be formatted when output to the screen or the printer. The formatting part of a word processor may be controlled by commands embedded in the text while editing. These control commands may include functions such as new page, new paragraph, centre heading, new line, font (character size) and tab control.

The third requirement was that the text may be stored on tape or disc to be reloaded for use at a later date.

BASIC PROBLEMS

Having decided on the requirements the next stage was to design the program. To ease the job of writing the program I decided to use BASIC as much as possible, only using Assembler where BASIC would be too slow. The first problem to solve in designing the program was how to store the text. The solution that immediately sprung to mind was the use of BASIC string variables. This was discarded because all BASIC string variables are lost when the BREAK key is hit by accident, maybe destroying a large amount of work.

The solution adopted involves storing the text between address &E00 and &5000. The BASIC program would normally be stored in this space but can be moved above &5000 by setting the value of PAGE = &5000 before loading the BASIC program. This area of memory is then reserved for use by the Word Processor program. The BASIC program requires a number

of pointers into this text area; these are stored in the BASIC static variables A% to Z% which are only changed when assigned to or when the machine is switched off.

The text area may be divided into two sections. The first section normally contains all the text typed in to date. The second section is only used while in insert mode when it contains the text after the insert position. When the insert is complete the second section of text is appended to the first section. The following variables are used as pointers to the text:

S% is used as a pointer to the start of the first section of text.

E% is used as a pointer to the end of the first section of text.

Z% is used as a pointer to the start of the second section of text.

PAGE is used as pointer to the end of the second section of text. PAGE is the BASIC variable used to define the start of a program.

After deciding on the requirements and the format of the text area I progressed to the design of the program. The requirements called for a number of functions. How were these functions to be selected by the user? One option was to have the functions selected by a menu. Another option was to write a routine to interpret the commands input by the user and select the appropriate function. The selected option makes the user directly call the prodedure that performs that option in the BASIC command mode; for example, typing PROCEDIT invokes the editor routines. This option was selected because it was considered the simplest (the work is done by the BASIC interpreter) and the shortest (no additional code for a command interpreter or a menu).

This option also allows the user to intersperse other commands between word processor commands, eg ∗CAT to position the tape before loading or saving a file. A disadvantage of this approach is that the procedure names may be difficult to remember. This can be overcome by programming the procedure calls on to the function keys and labelling the function keys.

PROGRAM STRUCTURE

The Word Processor consists of two programs, an initalisation program and a main program. The initialisation program assembles the machine code, programs the function keys, sets PAGE = &5000 and chains the main program.

The main program consists of six command procedures plus

Listing 1. The Word Processor initialisation program.

```
100REM"****************************
110REM"        WORD PROCESSOR
120REM"    INITIALISATION PROGRAM
130REM"       (c) J Ansell 1983
140REM"****************************
150*TV255
160*FX6,0
170MODE 7
180PRINT"WORD PROCESSOR INITIALISATION"
190PROCASM
200PRINT
210PRINT"KEY 0 is EDIT TEXT"
220*KEY 0 !OCLEAR:PROCEDIT!M
230PRINT"KEY 1 is FORMAT TEXT"
240*KEY 1 !OCLEAR:PROCFORMAT!M
250PRINT"KEY 2 is SAVE TEXT"
260*KEY 2 !OCLEAR:PROCSAVE!M
270PRINT"KEY 3 is LOAD TEXT"
280*KEY 3 !OCLEAR:PROCLOAD!M
290PRINT"KEY 4 is SEARCH AND REPLACE"
300*KEY 4 !OCLEAR:PROCSANDR!M
310PRINT"KEY 8 is RUN"
320*KEY 8 !ORUN!M
330PRINT"KEY 9 is CLEAR TEXT"
340*KEY 9 !OCLEAR:PROCINIT!M
350*KEY 10 "PAGE=&5000!MOLD!M"
360PRINT
370PAGE=&5000:CHAIN"RWP"
380DEFPROCASM
390REM"ASSEMBLE LIST & MOVE TEXT
400LOCALSA%,EA%,OSWRCH,I,SS,SE,STP,DS
410SA%=&70:REM"START ADDRESS
420EA%=&74:REM"END ADDRESSS
430OSWRCH=&FFEE:REM"WRITE CHARACTER
440FOR I=0 TO 2 STEP 2
450P%=&0D00
460[OPT I
470.LLOOP:LDY #0:LDA (SA%),Y
480JSR OSWRCH:INC SA%:BNE XNOINC
490INC SA%+1:.XNOINC:LDA SA%:CMP EA%
500BNE LLOOP:LDA SA%+1:CMP EA%+1
510BNE LLOOP:RTS
520]
530NEXT
540SS=&70:REM"SOURCE START
550SE=&74:REM"SOURCE END
560STP=&78:REM"STEP
570DE=&7C:REM"DEST END
580FOR I=0 TO 2 STEP 2
590P%=&0D20
600[OPT I
610.MSTART:LDX #0
620.MCONT:LDA (SS,X):STA (DE,X)
630CLC:LDA STP:ADC DE:STA DE
640LDA STP+1:ADC DE+1:STA DE+1
650CLC:LDA STP:ADC SS:STA SS
660LDA STP+1:ADC SS+1:STA SS+1
670LDA SS:CMP SE:BNE MCONT
680LDA SS+1:CMP SE+1:BNE MCONT
690LDA (SS,X):STA (DE,X):RTS
700]
710NEXT
720ENDPROC
```

Listing 2. The main Word Processor program.

```
100REM"****************************
110REM"
120REM"         WORD PROCESSOR
130REM"    (c) J Ansell   1983
140REM"
150REM"TESTED USING:-
160REM"
170REM" OPPERATING SYSTEM 1.2
180REM" EPSON MX-80F/T III PRINTER
190REM"
200REM"****************************
210MODE 7
220PROCINIT
230PRINT CHR$(133)+"WORD PROCESSOR READY"
240END
250DEF PROCINIT
260REM"INITIALISE EDIT VARIABLES
270REM"===================
280S%=&E02
290T%=0
300C%=0
310E%=S%
320Z%=PAGE-1
330?S%=32
340ENDPROC
350REM"****************************
360REM"START OF PROCEDURES USED
370REM"WHEN EDITING
380REM"****************************
390DEF PROCDEL
400REM"DELETE CHARACTER TYPED
410IF FNMCPOS=E%-1 THEN E%=E%-1:PROCLIST:ENDPROC
420IF FNMCPOS<E% THEN PROCDEL:ENDPROC
430IF E%=S% THEN ENDPROC
440VDU 127:E%=E%-1:C%=C%-1
450ENDPROC
460DEF PROCSINSERT
470REM"PROCEDURE TO START INSERT
480IF FNMCPOS=E% THEN ENDPROC
490IF Z%<>PAGE-1 THEN ENDPROC
500PROCMOVETEXT(E%,FNMCPOS,PAGE-1,-1)
510Z%=PAGE-E%+FNMCPOS-1
520E%=FNMCPOS
530PROCLIST
540ENDPROC
550DEF PROCMOVETEXT(P1%,P2%,P3%,P4%)
560REM"MOVE TEXT PROCEDURE
570REM"SEE ASSEMBLER
580!&70=P1%
590!&74=P2%
600!&78=P4%
610!&7C=P3%
620CALL &0D20
630ENDPROC
640DEF PROCLIST
650REM"LISTS TEXT ON TO SCREEN
660LOCAL SA%,EA%
670SA%=&70:EA%=&74
680CLS
690!SA%=T%*40+S%
700IF !SA%+999>E% THEN !EA%=E% ELSE !EA%=!SA%+999
710IF !SA%<>!EA% THEN CALL &0D00
720PRINT TAB(C%MOD40,C%DIV40);
730ENDPROC
740DEF PROCEINSERT
750REM"END INSERT PROCEDURE
760IF Z%=PAGE-1 THEN GOTO 820
770PROCMOVETEXT(Z%,PAGE-1,E%,1)
780E%=E%+PAGE-Z%-1
790Z%=PAGE-1
800PROCLIST
810ENDPROC
820IF KEY%=1 THEN GOTO 800 ELSE ENDPROC
830DEF PROCDEL
840REM"DELETE CHARACTER PROCEDURE
850LOCAL F%
860F%=FNMCPOS
870IF F%>=E% THEN ENDPROC
880PROCMOVETEXT(F%+1,E%,F%,1)
890E%=E%-1
900PROCLIST
910ENDPROC
920DEF PROCQUPPAGE
930IF C%>=0 THEN ENDPROC
940C%=C%+40:T%=T%-1:PROCLIST
950ENDPROC
960DEF PROCQDOWNPAGE
970IF C%<999 THEN ENDPROC
980C%=C%-40:T%=T%+1:PROCLIST
990ENDPROC
1000DEF PROCMOVECURSOR
1010REM"MOVE CURSOR PROCEDURE
1020PROCEINSERT
1030ON KEY%-135 GOTO 1040,1070,1100,1130
1040REM"LEFT CURSOR KEY
1050IF FNMCPOS>S% THEN C%=C%-1:VDU 8:PROCQUPPAGE
1060ENDPROC
1070REM"RIGHT CURSOR KEY
1080IF FNMCPOS<E% THEN C%=C%+1:VDU 9:PROCQDOWNPAGE
1090ENDPROC
1100REM"DOWN CURSOR KEY
1110IF FNMCPOS+40<=E% THEN C%=C%+40:VDU 10:PROCQDOWNPAGE
1120ENDPROC
1130REM"UP CURSOR KEY
1140IF FNMCPOS-40>=S% THEN C%=C%-40:VDU 11:PROCQUPPAGE
1150ENDPROC
1160DEF PROCCOPY
1170REM"COPIES TEXT INTO BUFFER
1180IF LEN(C$)<255 AND FNMCPOS<E% THEN C$=C$+CHR$(?FNMCPOS):KEY%=137:PROCMOVECU
RSOR ELSE VDU 7
1190ENDPROC
1200DEF PROCEDIT
1210REM"EDIT TEXT PROCEDURE
1220REM"===============
1230 LOCAL KEY%
1240*FX4,1
1250PROCEINSERT
1260T%=0
1270C$=""
1280IF E%-S%<999 THEN C%=E%-S% ELSE C%=999
1290PROCLIST
1300REPEAT
1310KEY%=GET
1320IF NOT (KEY%<136 OR KEY%>139) THEN PROCMOVECURSOR :GOTO 1490
1330IF KEY%=127 THEN PROCDEL:REM"DELETE
1340IF KEY%=9 THEN PROCSINSERT:REM"CONTROL I
1350IF KEY%=5 THEN PROCEINSERT:REM"CONTROL E
1360IF KEY%=12 THEN PROCLAST:REM"CONTROL L
1370IF KEY%=6 THEN PROCFIRST:REM"CONTROL F
1380IF KEY%=14 THEN PROCNEXT:REM"CONTROL N
1390IF KEY%=16 THEN PROCPREVIOUS:REM"CONTROL P
1400IF KEY%=1 THEN PROCSINSERT:!E%=C%:E%=E%+LEN(C$):PROCEINSERT:REM"CONTROL A
1410IF KEY%=3 THEN C$="":REM"CONTROL C
1420IF KEY%=135 THEN PROCCOPY:REM"COPY
1430IF KEY%<32 OR KEY%>126 THEN GOTO 1490
1440?FNMCPOS=KEY%
1450VDU KEY%
1460C%=C%+1
1470IF E%=FNMCPOS-1 THEN E%=E%+1
1480IF C%>=1000 THEN T%=T%+1:C%=960
1490UNTIL KEY%=15
1500CLS:*FX4,0
1510ENDPROC
1520DEF FNMCPOS:=T%*40+C%+S%
1530DEF PROCFIRST
1540REM"MOVE CURSOR TO START OF FIRST PAGE
1550T%=0
1560C%=0
1570PROCLIST
1580ENDPROC
1590DEF PROCLAST
1600REM"MOVE CURSOR TO END OF LAST PAGE
1610IF E%-S%<1000 THEN T%=0 ELSE T%=(E%-S%-1000)DIV40+1
1620C%=E%-S%-40*T%
1630PROCLIST
1640ENDPROC
1650DEF PROCNEXT
1660REM"LIST NEXT PAGE
1670T%=T%+20
1680IF FNMCPOS+999>=E% THEN PROCLAST:ENDPROC
1690PROCLIST
1700ENDPROC
1710DEF PROCPREVIOUS
1720REM"LIST PREVIOUS PAGE
1730T%=T%-20
1740IF T%<0 THEN PROCFIRST:ENDPROC
1750PROCLIST
1760ENDPROC
1770REM"****************************
1780REM"START OF PROCEDURES USED
1790REM"FOR SEARCH AND REPLACE
1800REM"****************************
1810DEF PROCSANDR
1820REM"SEARCH AND REPLACE PROCEDURE
1830REM"===============
1840LOCAL SA%,EA%,PO%
1850PRINT CHR$(134);"SEARCH STRING"
1860INPUT LINE S$
1870PRINT CHR$(134);"REPLACE STRING"
1880INPUT LINE R$
1890PRINT CHR$(134);"GLOBAL OR SELECTIVE (G/S)";
1900INPUT F$
1910IF (ASC(F$) OR 32)=103 THEN G%=TRUE ELSE G%=FALSE
1920FS%=ASC(S$)
1930LS%=LEN(S$)
1940PO%=S%
```

many subsidiary procedures. These command procedures provide the word processor functions as follows:
PROCEDIT(f0) for entering and editing text.
PROFORMAT(f1) for formatting the text to the VDU screen or printer.
PROCSAVE(f2) for saving a text file.
PROCLOAD(f3) for loading a text file.
PROCSANDR(f4) for performing search and replace functions.
PROCINIT(f9) for initialising the text area.

These command procedures are normally programmed on to the function keys of the computer by the initialisation program. If the program is run the text area is initialised and control is returned to the BASIC command.

PROCEDIT

Calling the procedure PROCEDIT has the effect of entering the edit mode. This is the mode normally used to enter and edit text. When the edit mode is entered the cursor is positioned at the end of the first page of text. If the text is shorter than one page then it is positioned at the end of the text.

The cursor control keys are used to position the cursor where text is to be changed or inserted. If an attempt is made to move the cursor off the top or the bottom of the screen then the screen is scrolled down or up respectively. This allows any part of the text stored to be displayed on the screen. The text is changed by typing new text over the old. To continue, move the cursor to the end of the text and start typing.

A number of control codes can be used to ease the editing of the text. These are:
DELETE KEY Delete character.
This key deletes the character at the cursor position. If the cursor is at the end of the displayed text the previous character is deleted.
CTRL-I Start insert.
This control code causes a change to insert mode. When the CTRL-I code is input all text after the cursor is removed from the screen and the new text can then be inserted. Insert mode is exited by typing CTRL-E or moving the cursor. When insert mode is exited the screen is updated with the remaining text.
CTRL-E End insert.
This control code causes an exit from edit mode.
CTRL-O Exit edit mode.
This control code causes an exit from edit mode.

CTRL-F First page.
This control code updates the screen with the first page of text. The cursor is positioned at the first character.
CTRL-L Last page.
This control code updates the screen with the last page of text. The cursor is positioned at the last character.
CTRL-N Next page.
This control code updates the screen with the next page of text.
CTRL-P Previous page.
This control code updates the screen with the previous page of text.
COPY KEY Copy text into buffer.
This key copies the character at the cursor position into the text buffer and advances the cursor. The text buffer has a maximum size of 255 characters. This key used in conjunction with CTRL-A and CTRL-C enables a portion of text to be duplicated or moved.
CTRL-A Insert text buffer.
This control code inserts the contents of the text buffer at the current cursor position.
CTRL-C Clear text buffer.
This control code clears the text buffer.

PROCFORMAT

PROCFORMAT asks a number of questions as follows:
PRINTER OR VDU (P/V)
If the VDU is selected then the formatted text is displayed to the screen a page at a time. The text is formatted 40 characters to a line using all the embedded control commands. Different character fonts are displayed on the screen in different colours and underlined characters are displayed as flashing characters. If the printer is selected then the formatted text is

output to the printer.
DRAFT COPY (Y/N)
If draft copy is selected then the text is printed normally. If not then all text printed is emphasised to produce a good copy.
DOUBLE LINE SPACING (Y/N)
If double line spacing is selected two newline characters are output at the end of each line.
SINGLE SHEET OR CONTINUOUS PAPER (S/C)
If single sheet paper is to be used select S; this instructs PROCFORMAT to wait for the Return key before continuing with the next page.
NO. OF LINES PER PAGE
Requests the maximum number of lines to be output per page. For the VDU a value of 23 is recommended. For A4 paper a value of 60 is recommended.
PAGE NUMBERING (Y/N)
If page numbering is selected then a page number is printed at the top right hand corner of each page.

START PAGE NUMBER
Requests the page number of the first page. Subsequent pages have the page number incremented. This enables large documents to be stored in several files each of which can be output individually with the correct page numbers.

The format of the text being displayed or printed may be controlled by control commands inserted into the text while editing. These commands consist of two characters, the first character being an ↑, while the second character indicates the command, eg ↑ p indicates a new paragraph to PROCFORMAT. The character after the ↑ may be in upper or lower case. The following format control commands may be used:

```
1950REPEAT
1960IF ?P%=FS% THEN PROCFIRSTEQ
1970P%=P%+1
1980UNTIL P%>E%-LS%
1990ENDPROC
2000DEF PROCFIRSTEQ
2010REM"FIRST CHARACTER EQUAL CHECK REMAINDER OF STRING
2020T$=""
2030FOR I%=P% TO P%+LS%-1
2040T$=T$+CHR$(?I%)
2050NEXT
2060IF T$<>S$ THEN ENDPROC
2070IF G% THEN PROCREPLACE:ENDPROC
2080REM"QUERY LIST
2090SA%=&70:EA%=&74
2100CLS
2110'SA%=((P%-S%)DIV 40-10)*40+S%
2120IF 'SA%<S% THEN 'SA%=S%
2130PO%=P%-'SA%
2140IF 'SA%+800>E% THEN 'EA%=E% ELSE 'EA%='SA%+800
2150IF 'SA%<>'EA% THEN CALL &D00
2160PRINT TAB(13,23);CHR$(134);"REPLACE(Y/N)"
2170PRINT TAB(PO%MOD40,PO%DIV40);
2180F$=GET$
2190IF F$="y"OR F$="Y" THEN PROCREPLACE
2200CLS
2210ENDPROC
2220DEF PROCREPLACE
2230REM"REPLACE STRING AT P%
2240IF P%+LS%=E% THEN $P%=R$:P%=P%+LEN(R$):E%=P%:ENDPROC
2250PROCMOVETEXT(E%,P%,PAGE-1,-1)
2260Z%=PAGE-E%+P%-1+LS%
2270E%=P%+LEN(R$)
2280$P%=R$
2290PROCMOVETEXT(Z%,PAGE-1,E%,1)
2300E%=E%+PAGE-Z%-1
2310Z%=PAGE-1
2320P%=P%+LEN(R$)-1
2330ENDPROC
2340REM"*****************************
2350REM"START OF PROCEDURES USED
2360REM"WHEN FORMATING TEXT
2370REM"*****************************
2380DEF PROCPRINT(S$)
2390REM"PRINT STRING TO PRINTER
2400LOCAL I%
2410S$=FNFONT(FONT%)+S$+CHR$(27)+"-"+CHR$(0)
2420VDU 2
2430FOR I%=1 TO LEN(S$)
2440VDU 1,ASC(MID$(S$,I%,1))
2450NEXT
2460VDU 3
2470ENDPROC
2480DEF PROCDISP(S$)
2490REM"PRINT STRING TO VDU
2500PRINT FNFONT2(FONT%)+S$;
2510ENDPROC
2520DEF PROCOUTSTR(S$)
2530REM"OUTPUT STRING ON PRINTER/VDU
2540S$=S$+CHR$(10)+CHR$(13)
2550IF O% THEN PROCPRINT(S$) ELSE PROCDISP(S$)
2560XLINE%=XLINE%+1
2570IF XLINE%>LPP% THEN PROCNEWPAGE
2580ENDPROC
2590DEF PROCNORMCH
2600REM"OUTPUT NORMAL CHARACTER
2610IF K%=94 THEN F%=TRUE ELSE PROCOUTCHAR
2620ENDPROC
2630DEF PROCOUTCHAR
2640REM"ADD CHARACTER TO OUTPUT STRING
2650IF K%=32 THEN B$="":B%=0 ELSE B$=B$+CHR$(K%):B%=B%+1
2660A$=A$+CHR$(K%):A%=A%+1
2670IF A%<W% THEN ENDPROC
2680PROCOUTSTR(LEFT$(A$,A%-B%))
2690A$=STRING$(M%," ")+B$:A%=B%+M%:B$="":B%=0
2700ENDPROC
2710DEF PROCCTRLCH
2720REM"OUTPUT CONTROL CHARACTER
2730F%=FALSE
2740IF K%=94 THEN PROCOUTCHAR:ENDPROC
2750K%=k% OR 32
2760IF K%<97 OR K%>122 THEN ENDPROC
2770ON K%-96 GOTO 2780,2790,2800,2810,2820,2830,2840,2850,2860,2870,2880,2890,2900,2910,2920,2930,2940,2950,2960,2970,2980,2990,3000,3010,3020,3030
2780PROCMARGIN(2):ENDPROC:A
2790PROCMARGIN(4):ENDPROC:B
2800PROCMARGIN(8):ENDPROC:C
2810PROCMARGIN(16):ENDPROC:D
2820M%=0:ENDPROC:E
2830ENDPROC:F
2840ENDPROC:G
2850ENDPROC:H
2860W%=W% 2:ENDPROC:I
2870W%=W%-4:ENDPROC:J
2880W%=W%-8:ENDPROC:K
2890W%=W%-16:ENDPROC:L
2900PROCOUTSTR(STRING$((W% A%)/2," ")+A$):PROCCL:ENDPROC:M
2910PROCOUTSTR(A$):PROCCL:ENDPROC:N
2920ENDPROC:O
2930PROCOUTSTR(A$):PROCOUTSTR(""):PROCCL:K%=32:PROCNORMCH:PROCNORMCH:ENDPROC:P
2940PROCOUTSTR(A$):PROCCL:PROCNEWPAGE:ENDPROC:Q
2950PROCOUTSTR(STRING$(W% A%," ")+A$):PROCCL:ENDPROC:R
2960ENDPROC:S
2970ENDPROC:T
2980IF O% THEN A$=A$+CHR$(27)+"-"+CHR$(1):ENDPROC ELSE A$=A$+CHR$(136):A%=A%+1:ENDPROC:U
2990IF O% THEN A$=A$+CHR$(27)+"-"+CHR$(0):ENDPROC ELSE A$=A$+CHR$(137):A%=A%+1:ENDPROC:V
3000PROCWI(37):FONT%=1:ENDPROC:W
3010PROCWI(63):FONT%=2:ENDPROC:X
3020PROCWI(73):FONT%=3:ENDPROC:Y
3030PROCWI(126):FONT%=4:ENDPROC:Z
3040DEF PROCOUTCH
3050REM"OUTPUT CHARACTER IN K%
3060IF F% THEN PROCCTRLCH ELSE PROCNORMCH
3070ENDPROC
3080DEF PROCFORMAT
3090REM"TEXT FORMATING PROCEDURE
3100REM"======================
3110LOCAL I%,FONT%,XLINE%,LPP%,PNO%,SI%
3120KEY%=0
3130PROCEINSERT
3140SI%=TRUE
3150CLS:INPUT "PRINTER OR VDU (P/V)"F$
3160IF (ASC(F$) OR 32)=112 THEN O%=TRUE:PROCMOREQ ELSE O%=FALSE
3170FS$=""
3180A$=STRING$(255," "):B$=STRING$(255," ")
3190PROCWI(73):FONT%=3
3200INPUT "NO. OF LINES PER PAGE "LPP%
3210INPUT"PAGE NUMBERING (Y/N)"F$
3220IF (ASC(F$) OR 32)=121 THEN N%=TRUE ELSE N%=FALSE
3230IF N% THEN INPUT"START PAGE NUMBER "PNO%
3240PNO%=PNO%-1:PROCNEWPAGE
3250PROCCL:F%=FALSE
3260FOR I%=S% TO E%
3270K%=?I%:PROCOUTCH
3280NEXT
3290ENDPROC
3300DEF PROCMOREQ
3310REM"REQUESTS PRINT PARAMETERS
3320INPUT"DRAFT COPY (Y/N)"F$
3330IF (ASC(F$) OR 32)=110 THEN D%=4 ELSE D%=0
3340IF (ASC(F$) OR 32)=121 THEN D$=CHR$(10) ELSE D$=""
3350IF (ASC(F$) OR 32)=121 THEN D$=CHR$(10) ELSE D$=""
3360INPUT"SINGLE SHEET OR CONTINUOUS PAPER(S/C)"F$
3370IF (ASC(F$) OR 32)=115 THEN SI%=TRUE ELSE SI%=FALSE
3380ENDPROC
3390DEF PROCCL
3400REM"CLEAR LINE
3410A$=STRING$(M%," "):B$="":A%=M%:B%=0
3420ENDPROC
3430DEF PROCWI(NUM%)
3440REM"SETS W% TO WIDTH
3450IF O% THEN W%=NUM% ELSE W%=38
3460M%=0
3470IF LEN(A$)<>255 THEN PROCOUTSTR(A$)
3480PROCCL
3490ENDPROC
3500DEF FNFONT(NUM%)
3510REM"ANSWER IS SELECTED PRINTER FONT
3520ON NUM%+D% GOTO 3530,3540,3550,3560,3570,3580,3590,3600
3530=D$+CHR$(14)+CHR$(18)+CHR$(27)+"F    "
3540=D$+CHR$(14)+CHR$(15)+CHR$(27)+"F    "
3550=D$+CHR$(20)+CHR$(18)+CHR$(27)+"F    "
3560=D$+CHR$(15)+CHR$(20)+CHR$(27)+"F    "
3570=D$+CHR$(14)+CHR$(18)+CHR$(27)+"E    "
3580=D$+CHR$(14)+CHR$(15)+CHR$(27)+"F    "
3590=D$+CHR$(20)+CHR$(18)+CHR$(27)+"E    "
3600=D$+CHR$(15)+CHR$(20)+CHR$(27)+"F    "
3610DEF PROCMARGIN(P1%)
3620REM"SETUP MARGIN
3630M%=M%+P1%
3640IF M%>A% THEN A$=A$+STRING$(M%-A%," "):A%=M%
3650ENDPROC
3660DEF PROCNEWPAGE
3670REM"PRINT PAGE HEADINGS
3680LOCAL FT%,C$
3690VDU 7
3700IF SI% THEN INPUT'"PRESS RETURN FOR NEXT PAGE"C$ ELSE PROCPRINT(CHR$(12))
3710IF O% THEN PROCPRINT(CHR$(27)+"8"+CHR$(13))
3720FT%=FONT%:FONT%=3
3730PNO%=PNO%+1
3740XLINE%=0:C$="Sheet   "+STR$(PNO%)
3750IF N% THEN PROCOUTSTR(STRING$(73-LEN(C$)," ")+C$)
3760PROCOUTSTR("")
3770FONT%=FT%
3780ENDPROC
3790DEF FNFONT2(NUM%)
3800REM"ANSWER IS SELECTED VDU FONT
3810ON NUM% GOTO 3820,3830,3840,3850
3820=CHR$130
3830=CHR$131
3840=CHR$135
3850=CHR$134
3860REM"*****************************
3870REM"START OF PROCEDURES USED
3880REM"WHEN SAVING AND LOADING TEXT
3890REM"*****************************
3900DEF PROCSAVE
3910REM"SAVE TEXT INTO FILE
3920REM"=================
3930KEY%=0
3940PROCEINSERT
3950CLS
3960INPUT "FILE NAME TO BE SAVED ?"F$
3970*OPT 1,1
3980CH%=OPENOUT(F$)
3990FOR I%=S% TO E%
4000BPUT#CH%,?I%
4010NEXT
4020CLOSE#CH%
4030ENDPROC
4040DEF PROCLOAD
4050REM"LOAD TEXT FROM FILE
4060REM"=================
4070CLS
4080INPUT "FILE NAME TO BE LOADED ?"F$
4090*OPT 1,1
4100CH%=OPENIN(F$)
4110REPEAT
4120?E%=BGET#CH%
4130E%=E%+1
4140UNTIL EOF#CH%
4150CLOSE#CH%
4160E%=E%-1
4170ENDPROC
```

↑ **a** moves the left margin two characters to the right. All lines of text after this control command have two additional spaces inserted at the start of the line. If the current display or print position is to the left of the margin then spaces are inserted up to the left of the margin then spaces are inserted up to the margin, ie it has a similar effect to a tab character. The effect of this command is cumulative, so ↑ a ↑ a moves the left margin four characters to the right, for example.

↑ **b** moves the left margin four characters to the right. Works in a similar way to ↑ a.

↑ **c** moves the left margin eight characters to the right. Works in a similar way to ↑ a.

↑ **d** moves the left margin 16 characters to the right. Works in a similar way to ↑ a.

↑ **e** resets the left margin to the start of a line, resetting the cumulative effect of all previous ↑ a, ↑ b, ↑ c and ↑ d commands.

↑ **i** decreases the number of characters per line used by

PROCFORMAT by two. The effect of this command is cumulative, so ↑ i ↑ i decreases the number of characters per line by four, for example.

↑ **j** decreases the number of characters per line by four. Works in a similar way to ↑ i.

↑ **k** decreases the number of characters per line by eight. Works in a similar way to ↑ i.

↑ **l** decreases the number of characters per line by 16. Works in a similar way to ↑ i.

↑**m** is used to print or display text ▶

in the middle of a line. The text affected starts after the preceding new line and ends with ↑ m. Subsequent text is displayed or printed on a new line. This is useful for headings.

↑ **n** forces a new line.

↑ **p** forces a new paragraph. A blank line is displayed or printed between paragraphs. The first line of a paragraph is indented by two spaces.

↑ **q** forces a new page.

↑ **r** is used to print or display text to the right of a line. The text affected starts after the preceding new line and ends with ↑ r. Subsequent text is displayed or printed on a new line. This is useful for printing the right hand address on a letter.

↑ **u** switches on underline printing. All text after ↑ u is underlined until an ↑ v or the next new line. Underlined text is displayed as flashing text on the VDU.

↑ **v** switches off underlined printing.

↑ **w** forces a new line and sets subsequent printing to 37 enlarged characters to a line on the printer. Text displayed on the VDU after ↑ w is coloured green. This control command also resets the effect of ↑ a, ↑ b, ↑ c, ↑ d, ↑ i, ↑ j, ↑ k, and ↑ l.

↑ **x** has a similar effect to ↑ w except that subsequent printing is 63 enlarged condensed characters per line and the text displayed on the VDU is coloured yellow.

↑ **y** has a similar effect to ↑ w except that subsequent printing is 73 normal characters per line and the text displayed to the VDU is coloured white.

↑ **z** has a similar effect to ↑ w except that subsequent printing is 126 condensed characters per line and the text displayed to the VDU is coloured cyan.

PROCSAVE

PROCSAVE asks for a name for the file to be saved. It then saves the text area into the file specified.

PROCLOAD

PROCLOAD asks for the name of the text file to be loaded. If

PROCLOAD is called while in insert mode in PROCEDIT then the file is inserted at the cursor position, otherwise the file is appended to the current text area.

If a new file is required the text area must be cleared using PROCINIT before loading the file.

PROCSANDR

PROCSANDR searches for a given string and replaces it with a new string. It prompts the user as follows:

SEARCH STRING Type in string to be changed.

REPLACE STRING Type in new string.

GLOBAL OR SELECTIVE (G/S) If G then all occurrences of the search string are changed. If S then each time the search string is found the page of text is displayed on the screen and the user is asked:

REPLACE (Y/N) The string is replaced if the answer is Y.

PROCINIT

PROCINIT is called to initialise the text area when a new file is to be started in the same way as NEW is used before typing in a new BASIC program.

Simon Goodwin

THE ZX LPRINT INTERFACE

Sinclair's printer is cheap, convenient and small, but not exactly letter quality. This Centronics interface will let you get better printouts.

When Sinclair first announced the ZX Spectrum computer it was promoted as a machine for games and learning about computers. Since then it has found use as a business planner, a filing system and even as a word-processor. The system was designed to work with only one printer — the cheap but restricted ZX printer made by Sinclair. Unlike other home computers the Spectrum does not contain the necessary interface electronics to allow it to be connected to a more sophisticated printer.

The ZX LPRINT is a device intended to change all that. When LPRINT is connected the Spectrum can communicate with almost any popular printer. LPRINT enables the Spectrum to generate signals in the Centronics format, one of the standard ways of allowing a computer to communicate with a printer.

THE HARDWARE

The ZX LPRINT is a small black box, about 7 cm on a side and 2.5 cm deep. It plugs into the back of the computer where the Sinclair printer would normally fit. A connector on the opposite side of the box to the

computer accommodates a standard printer lead, similar to the one used on the BBC Micro and other home computers.

Simple instructions are neatly printed on the underside of the interface box — that way you can't lose them! The only mechanical flaw of the design is that it does not provide a socket for further devices which need access to the Spectrum's edge connector — once you've plugged in the ZX LPRINT you cannot connect anything else. The makers plan to produce a buffer to allow the ZX LPRINT to be used at the same time as Microdrives.

ALTERNATIVES

There have been other Centronics add-ons produced for the Spectrum, but the ZX LPRINT has features which mark it out from the alternatives.

When you use the LPRINT or LLIST commands on a normal Spectrum the computer assumes that a Sinclair printer is connected, and sends a sequence of on-off signals to the expansion connector. Any message may be written by telling the computer whether or not to make a mark as it passes each part of the

paper.

The Sinclair printer is simple and effective, but it is also rather slow and the print quality can be poor. More expensive printers communicate using numeric codes rather than an on-off signal. When a character is to be printed a number is transferred between the computer and the printer, and the value determines the character which is output. The most popular code is called ASCII, and ASCII code is indeed used inside the Spectrum, but not for communication with the Spectrum printer. A Centronics interface allows ASCII codes to be sent to a printer.

It is a fairly simple feat of electronic wizardry to graft a Centronics interface onto a Spectrum. The snag is that the computer does not have any way of knowing that a quality printer is connected. If you type LLIST the computer will send the on-off sequence expected by a Sinclair printer, and not the ASCII which the new interface needs.

Most interface suppliers get around this difficulty by supplying a tape which must be loaded whenever a quality printer is to be used. The tape contains instructions which tell the computer to use the interface rather than try to communicate with the Sinclair printer. This approach works, but it has the snag that you must always load the tape before you can do any printing. The added printer instructions also use up some of your computer's memory, which may mean that a few programs won't work.

A NEW SOLUTION

The ZX LPRINT provides a much neater solution. As soon as the interface is connected some of the instructions built into the Spectrum become unnecessary. These are the instructions which tell the computer how to control the Sinclair printer. The ZX LPRINT contains electronics which temporarily replaces those instructions with new ones to work the Centronics interface. Unless you examine the Spectrum ROM in great detail you cannot even tell that this has happened.

The effect is that, from the moment you connect the ZX LPRINT, the computer communicates with your quality printer instead of the Sinclair one. You may use the LPRINT and LLIST commands just as you would normally, except that you get a quality printout.

LIMITATIONS

The ZX LPRINT has a few limitations. You cannot send user-defined graphics characters to the printer, ▶

A close up of the ZX LPRINT from Euroelectronics. It's neat and compact but the edge connector PCB tracks are exposed (don't drop paperclips down the back of the Spectrum) and doesn't allow other peripherals to be connected — yet.

because different makes of printer have different ways of showing graphics. Alphabetic characters or symbols are standardised, but if you try to LPRINT or LLIST a picture made up of user-defined graphics they do not appear on the printer.

The ZX LPRINT also sends a Carriage Return and Line End at the end of each line. Many modern printers don't need the Line Feed signal — they automatically wind on the paper as they move the print head back. If you use the ZX LPRINT on one such printer you may find that text is printed double-spaced, with a blank line between each line of text. The instructions don't mention this problem. You have to alter the setting of the printer (usually by an internal switch) so that the Line Feed is ignored: you can't stop the ZX LPRINT sending the Line Feed.

OUT OF CONTROLS

Carriage Return and Line Feed are just two of the control codes recognised by most printers. Other control codes vary from one printer to the next, allowing you to select special features of the printer such as different sized letters, graphics characters and so forth. It can be difficult to send these codes to a printer since the Spectrum has its own control codes, used to select colours, keywords and so on.

Your printer might treat CHR$(190) as a graphics control code, but if you try to PRINT or LPRINT that code the word PEEK will appear, because the Spectrum uses codes greater than 164 to represent keywords.

The ZX LPRINT allows you to

send a special character which 'turns off' the Spectrum's recognition of codes greater than 127, so that you can send many control codes to the printer without the computer changing them en route. Unfortunately ZX LPRINT does not stop the computer trying to convert codes less than 32. These values are also used as control codes so that some values are still difficult to transmit.

You must remember to 'turn on' the Spectrum's recognition of codes greater than 127 after you have turned it off, or listings will be gibberish — none of the keywords will appear.

COPING WITH COPY

So far we have only discussed the commands LLIST and LPRINT. The Spectrum also has a COPY command, which sends a black-and-white copy of the screen picture to the Sinclair printer. We noted earlier that graphics codes vary from one printer to the next, so that ZX LPRINT cannot be expected to send graphic pictures to all kinds of printer. The best solution is a compromise.

The ZX LPRINT does not contain instructions to process a COPY command, but it can pass the command

on to a machine code program stored in the Spectrum's memory. The makers of ZX LPRINT sell programs to carry out the COPY function on most popular printers. You have to load a tape to use COPY, but many kinds of printer may be used and you can still use LPRINT and LLIST without the need for a tape.

For the purposes of this review we intended to show an example of COPY through the ZX LPRINT. Unfortunately the author uses an ageing Epson MX80 Type I printer, which only has very simple graphics facilities. COPY machine-code is not available to work it. However, it is still possible to achieve the effect of COPY from BASIC. The picture in Fig. 1 was printed using the ZX LPRINT and the short BASIC program used to produce it is shown in Fig. 2.

If you have an MX80 Type I you can use the program to produce a (rather slow) COPY of whatever is on the screen. The picture will be 'stretched' since the printer generates oblong dots rather than square ones. The program in Fig. 2 also works on an OKI Microline 80 printer if you make minor changes — put 29 instead of 15 in line 100, and replace 160 by 128 in line 140.

FINAL REPORT

ZX LPRINT is a remarkable device which packs considerable ingenuity into a small box. It costs £41.40 including VAT, plus a rather high charge of £12.08 for a ready-made cable. The COPY software is available for most popular printers — each version comes with a demonstration and costs £5. The interface is made by Euroelectronics, Montpellier Retreat, Cheltenham, Gloucestershire GL50 2XG.
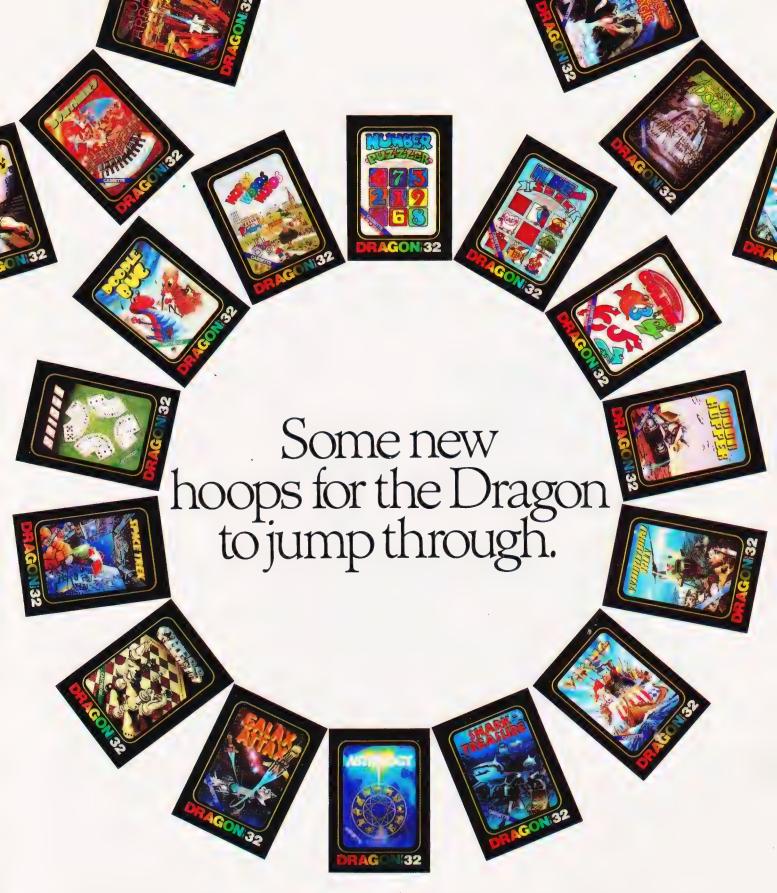
At these prices ZX LPRINT is an expensive add-on for a Spectrum, but its price is reasonable considering the clever design and the cost of a printer to connect to it. If you've got a Spectrum and you're sick of 'space-age' Sinclair printouts the interface might be a good investment. If your Spectrum is a second computer and you already have a quality printer then the ZX LPRINT should be almost irresistible.

Fig. 1 An example of the Epson MX80 I version of COPY.

```
 60 REM ***********************     C,R)+POINT (C+1,R) *2+POINT (C,R-
 70 REM * MX80 Type I Printer *     1) *4+POINT (C+1,R-1) *8+POINT (C,
 80 REM * ZX Spectrum COPY V2 *     R-2) *16+POINT (C+1,R-2) *32)
 90 REM ***********************     150 NEXT C
100 LPRINT CHR$ 5;CHR$ 15            160 LPRINT A$
110 FOR R=175 TO 0 STEP -3           170 NEXT R
120 LET A$=""                        180 LPRINT CHR$ 4
130 FOR C=0 TO 255 STEP 2            190 STOP
140 LET A$=A$+CHR$ (160+POINT (
```

Fig. 2 The COPY program that generated Fig. 1, printed out using itself!

Some new hoops for the Dragon to jump through.

Unless you can already write software programs, the usefulness of a home computer is governed, to a great extent, not by its power but by the software available.

After all what's the use of a very large memory if you can't use it properly?

So at Dragon we not only took great care when designing the actual computer, we also made sure that the people who designed the software took equal care.

By using only top writers, and by giving them subjects of interest to you, we managed to produce some of the very best programs around.

Take, for example, the Cyrus Chess program. Designed by a grand master it, not surprisingly, won the accolade of Microcomputer Chess Champion.

Of course you don't have to be a grand master to play it, but, on level nine, you do need to be one to beat it.

Mind you, that's just one of nearly a hundred Dragon software titles available and, though they're all fun, they're certainly not all games.

There are programs that can teach your children spelling and mathematics.

Another can teach you how to type.

And, although some of our other titles are displayed above, the best way of seeing our full range of software isn't by looking at this ad.

It's by going down to your local Dragon stockist.

DRAGON

David Bright M.Sc, Ph.D

RESCUING SOURCE CODE

When developing Assembler language programs on a cassette-based system, it's tempting not to back up the source code. If it crashes, this program may provide salvation.

Are you one of those unfortunates trying to develop assembler language programs on a TRS-80 or Genie cassette-based system? If so, then you know that awful moment when a program under test runs wild. After a short pause, during which the rogue program wreaks havoc with your system, a hang-up or spontaneous re-boot is obtained, and you realise with horror that you made your latest cassette copy of your source code more than two hours ago.

With trembling hands you try to restart your Editor/Assembler program. Another hang-up or spontaneous re-boot. Two hours program development effort down the drain again.

If you suffer from this syndrome, then cheer up. RESCUE is at hand. This article describes RESCUE, a utility program that can recover your source code in the form of an EDTASM-compatible cassette file that can be loaded by a fresh copy of EDTASM.

The utility works in conjunction with both Tandy's EDTASM, and Microsoft's EDTASM-PLUS and has proven to be an invaluable aid in minimising lost time in program development.

BACKGROUND

I have owned my Genie (original model) for a little over a year now. Unfortunately, the family budget does not (yet) allow me the luxury of a disc unit. As my primary interests in home computing are in music and graphics applications, I work mainly with machine language. After fooling around for a few weeks POKEing machine

language programs into place, I decided that I was wasting my time, and acquired Tandy's EDTASM Editor/Assembler program. This was a big improvement, but I discovered that I now spent most of my evenings watching the cassette player run. I found myself repeating the cycle:
1) Enter source program, and edit it.
2) Write two copies of the source program to tape.
3) Assemble the program, and write the executable version to tape.
4) Load the executable version.
5) Try a test run.
6) Reload the EDTASM program.
7) Reload the saved source code.
8) Go to 1.

As can be imagined, progress was extremely slow. Yet, if I omitted any of these steps, the chances were that no progress would be made at all, because I would lose my source code, and have no backup copy. There had to be a better way.

Fortunately, this period did not last long, and after a memory expansion to 48K, I graduated to Microsoft's Editor/Assembler-Plus. Now I had sufficient memory to accommodate both source and executable versions of the program simultaneously, and by using the EDTASM-PLUS assemble-to-memory facility, I could skip steps 2,3,4,5, and 7 of my development cycle, and proceed directly from assembly to testing under the built-in Z-BUG monitor. Finally, program development began to proceed at a reasonable pace. By using the Z-BUG breakpoint facility I could keep control of the program and prevent runaways that could corrupt my system.

However, things were not yet perfect. Although Z-BUG offers superb debug and execution control facilities, its use implies the acceptance of a number of restrictions.

A minor restriction is that the Break key may not be used as part of your test program's input, as this interrupts the test program, and returns control to Z-BUG. This is a deterrent to creation of user-friendly programs, since one of the first things a user learns is that the Break key is the universal 'get-out' when things are not going the way they should. Nevertheless, this is easily handled with only a little inconvenience. During testing, I use the '@' key in place of the Break key, and substitute the use of the Break key in the final production version.

A much more serious restriction is that Z-BUG does not

allow step-by-step tracing of instructions in ROM. If ROM routines are used, a breakpoint must be set immediately after the CALL, and when single-stepping one must be careful to enter continuous running mode before the CALL to ROM is encountered, otherwise Z-BUG will refuse to continue tracing.

Most annoying of all, Z-BUG redefines the action of several RST instructions that are essential to the use of the ROM arithmetic routines.

In the face of these restrictions, I tend to simulate ROM routines in the initial checkout phases, and do the later testing without the protection of Z-BUG. As a result, roughly once a week, something would go wrong, my program would run wild, and leave me with a corrupted source code buffer, a damaged copy of the EDTASM-PLUS program, or both.

Corrupted text in the source code buffer does not present a serious difficulty, since the next attempt to assemble reveals the problem, and the fault can be corrected with the Editor. Damage to source code line numbers or to the buffer control characters is more serious, but this is again immediately detected by the assembler, and can usually be corrected by renumbering, or by deletion of one or more source code lines.

Corruption of the EDTASM program itself is usually disastrous, since on attempting to restart EDTASM, the system either performs a spontaneous reboot, or just hangs up and apparently does nothing. In this situation, EDTASM simply has to be reloaded, with consequent loss of the source code buffer contents.

One evening, there I was, for the umpteenth time, with a damaged EDTASM resident in memory, and the only cassette copy of the source code was one I had made after my last development session the previous evening. I had made some major changes to my application program, and my source code, representing the last two hours work, was still sitting in the memory. Inspection with a monitor showed that it was apparently undamaged, but I had no way of recovering it in a useful format.

I knew that simply reloading EDTASM-PLUS would be of no avail, since the Microsoft copyright message, which is printed on initial start-up, plus some initialisation routines occupy the source code buffer area on initial load, and would therefore overwrite my precious code. I therefore used a monitor to dump the buffer area to a 'SYSTEM' format tape, reloaded EDTASM-PLUS, allowed it to initialise itself, and reloaded the buffer dump. This was still no good. Attempts to write out the buffer resulted in EDTASM-PLUS steadfastly declaring 'BUFFER EMPTY'. I went back to my monitor, and attempted to patch the EDTASM-PLUS internal pointers, with the result that I again produced a useless copy of EDTASM-PLUS.

I resolved to do something about this situation once and for all. What was required was a program that could be used as a matter of routine to recover the source buffer in this frequently occurring situation. I set my dump tape aside, and set to work. This EDTASM RESCUE program is the result.

PROGRAM DESCRIPTION

Specifically, the program offers the following features:
● The ability to list an EDTASM source buffer in order to check that it has not been corrupted, or at least to locate a corrupted portion of the buffer.
● The ability to produce a listing of an EDTASM source buffer on a printer.
● The ability to inspect an EDTASM buffer, byte by byte, to diagnose corruption problems, and the ability to edit the buffer to correct corruption. The memory inspection facility displays the memory in both ASCII and Hexadecimal representations.
● The ability to create a cassette tape of the buffer which can be loaded by a freshly loaded copy of EDTASM.

The program works with Microsoft's EDTASM-PLUS, and with Tandy's EDTASM. The only thing which needs to be adapted is the pointer to the starting location of the buffer. This may vary from version to version.

According to John T. Blair ("Custom EDTASM", *80 Microcomputing*, August 1980) the starting location of Tandy's EDTASM source buffer is at 5CF9 Hex, but in the version I have, the buffer starts at 5CF0 Hex.

In EDTASM-PLUS, the source code buffer normally starts at 7221 Hex. If the 'Quash' command has been used to eliminate Z-BUG and/or the Assembler, then this will be at a lower location. If a working version of the program is created using the normal address of the buffer, this can be changed at run time to accommodate use of 'Quash' by means of the built-in memory edit facility.

If you are in doubt as to where your version of EDTASM keeps its source code buffer, load EDTASM and enter the text 'HELLO' as source line 100. Then exit from EDTASM, load either a monitor program or the RESCUE program assembled as listed in Fig. 1, and inspect the memory until the byte string 'HELLO' is found. This string should be preceded by three bytes with hex values 64 00 05. The location containing value 64 is the start of the buffer. Enter this value in the source of the RESCUE program and assemble.

HOW IT WORKS

The format of each line of EDTASM source code in memory is as follows:
Bytes 0-1: Line number, stored as a binary integer. Thus byte 0 contains the least significant byte (LSB) and byte 1 contains the most significant byte (MSB).
Byte 2 contains a count of text characters in the line.
Bytes 3-257 contain the text of the line. Tabs are represented as a single character (09). Carriage return characters are not stored.
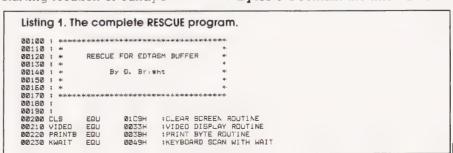
The buffer is terminated by a dummy line number, 0FFFF Hex.

The format on tape is rather different. The tape starts with the usual leader of 255 zeroes followed by the A5 Hex synchronisation byte. This is followed by a D3 Hex label byte which identifies the file as an EDTASM source file, and the label byte is followed by a six-character file name.

Each source code line follows in the following format:
Bytes 0-4 contain the line number

Listing 1. The complete RESCUE program.

```
00100 ; *********************************
00110 ; *                               *
00120 ; *    RESCUE FOR EDTASM BUFFER   *
00130 ; *                               *
00140 ; *        By D. Bright           *
00150 ; *                               *
00160 ; *                               *
00170 ; *********************************
00180 ;
00190 ;
00200 CLS     EQU     01C9H   ;CLEAR SCREEN ROUTINE
00210 VIDEO   EQU     0033H   ;VIDEO DISPLAY ROUTINE
00220 PRINTB  EQU     003BH   ;PRINT BYTE ROUTINE
00230 KWAIT   EQU     0049H   ;KEYBOARD SCAN WITH WAIT
```

```
00240 CASOFF   EQU    01F9H        ;CASSETTE SHUT OFF ROUTINE
00250 CASSON   EQU    0212H        ;CASSETTE STARTUP ROUTINE
00260 LEADER   EQU    0284H        ;WRITE CASSETTE LEADER ROUTINE
00270 TAPEWR   EQU    0264H        ;CASSETTE WRITE ROUTINE
00280 PAUSE    EQU    0060H        ;DELAY LOOP ROUTINE
00290 ;
00300 ;*****************************************************
00310 ;
00320 ;        ORG    0FC00H       ;FOR 48K RAM SYSTEM
00330 ;
00340 ;*****************************************************
00350 ;
00360 START    LD     SP,START            ;INITIALISE STACK POINTER
00370          DI                  ;DISABLE INTERRUPTS
00380          CALL   CLS          ;CLEAR SCREEN
00390          LD     A,15
00400          CALL   VIDEO        ;TURN OFF CURSOR
00410          LD     HL,MSG       ;PUT OUT SIGN ON MSG
00420          CALL   OUT
00430          CALL   GETKEY       ;GET USER RESPONSE
00440          CP     'Q'          ;QUIT
00450          JP     Z,0000H      ;REBOOT
00460          CP     'L'          ;LIST
00470          JR     Z,LIST
00480          CP     'W'          ;WRITE TAPE
00490          JR     Z,PUNCH
00500          CP     'P'          ;PRINT
00510          JR     Z,PRINT
00520          CP     'E'          ;EDIT MEMORY
00530          JP     Z,EDIT
00540          JR     START        ;BAD INPUT - TRY AGAIN
00550 ;
00560 OUT      LD     A,(HL)       ;PRINT MESSAGE POINTED TO BY HL
00570          OR     A            ;AND TERMINATED BY ZERO BYTE
00580          RET    Z
00590          PUSH   HL
00600          CALL   VIDEO        ;DISPLAY BYTE IN 'A' REGISTER
00610          POP    HL
00620          INC    HL
00630          JR     OUT
00640 ;
00650 ; LIST EDTASM SOURCE BUFFER ON VIDEO SCREEN
00660 ;
00670 LIST     CALL   CLS          ;CLEAR SCREEN
00680          LD     HL,LISTR     ;LIST BUFFER ON VDU
00690          LD     (WRITER),HL
00700          LD     HL,VIDEO     ;PLANT VIDEO DISPLAY
00710                              ;ROUTINE
00720          LD     (OUTR),HL
00730          LD     HL,1000H     ;SET DELAY TO SLOW DOWN
00740          LD     (WAIT),HL    ;THE LISTING
00750          LD     A,1          ;SET HALT MODE ON
00760          LD     (HALT),A
00770          LD     DE,(BUF)     ;EDTASM BUFFER START ADDRESS
00780          CALL   ADDRES       ;PUT OUT ADDRESS OF FIRST LINE
00790          LD     A,5          ;INITIALISE CURSOR POSN
00800          LD     (CURSOR),A
00810          JR     GO
00820 ;
00830 ; SEND EDTASM BUFFER LISTING TO PRINTER
00840 ;
00850 PRINT    LD     HL,LISTR     ;PRINT ROUTINE
00860          LD     (WRITER),HL
00870          LD     HL,PRINTB    ;PLANT PRINT BYTE ROUTINE
00880          LD     (OUTR),HL
00890          LD     HL,0001      ;VERY SHORT DELAY
00900          LD     (WAIT),HL    ;YOU WON'T NOTICE IT
00910          XOR    A            ;SET HALT MODE OFF
00920          LD     (HALT),A
00930          LD     DE,(BUF)     ;EDTASM BUFFER START
00940          CALL   ADDRES       ;PUT OUT FIRST LINE ADDRESS
00950          LD     A,5          ;INITIALISE CURSOR POSN
00960          LD     (CURSOR),A
00970          JR     GO
00980 ;
00990 ; WRITE EDTASM COMPATIBLE SOURCE TAPE
01000 ;
01010 PUNCH    CALL   CLS          ;CLEAR SCREEN
01020          CALL   GETNAM       ;GET FILE NAME
01030          LD     HL,READYC    ;PUT OUT READY CASSETTE
01040          CALL   OUT
01050          CALL   GETKEY       ;WAIT FOR KEYBOARD INPUT
01060          LD     HL,TAPEWR    ;PLANT CASSETTE WRITE
01070          LD     (WRITER),HL  ;ROUTINE
01080          XOR    A            ;SELECT INTERNAL CASSETTE
01090          CALL   CASSON       ;START MOTOR
01100          CALL   LEADER       ;WRITE LEADER
01110          LD     A,0D3H       ;EDTASM SOURCE TAPE LABEL
01120          CALL   WRITE
01130          LD     DE,NAME      ;PUT OUT FILE NAME (6 CHARS)
01140          LD     B,E          ;FROM NAME BUFFER
01150 LABEL    LD     A,(DE)
01160          INC    DE
01170          CALL   WRITE
01180          DJNZ   LABEL
01190 ;
01200 ; LIST BUFFER TO PRINTER/VIDEO/CASSETTE
01210 ;
01220 GO       LD     DE,(BUF)     ;START OF EDTASM BUFFER
01230 LINE     LD     A,(DE)       ;GET LINE NUMBER IN HL
01240          LD     L,A
01250          INC    DE
01260          LD     A,(DE)
01270          LD     H,A
01280          INC    DE
01290          LD     A,H          ;TEST FOR END OF EDTASM BUFFER
01300          AND    L
01310          CP     0FFH         ;LINE NUMBER IS 0FFFFH ?
01320          JR     Z,END        ;IF YES, ALL DONE
01330          CALL   ASCII        ;CONVERT LINE # FROM BINARY TO ASCII
01340          LD     A,32         ;PUT OUT A SPACE
01350          CALL   WRITE
01360          LD     A,(DE)       ;GET TEXT LENGTH
01370          INC    DE           ;BUMP POINTER
01380          OR     A            ;LOOK OUT FOR NULL LINES!
01390          JR     Z,NEXT
01400          LD     B,A          ;NUMBER OF CHARS TO 'B' REGISTER
01410 TEXT     LD     A,(DE)       ;PUT OUT TEXT
01420          CALL   WRITE
01430          INC    DE
01440          DJNZ   TEXT         ;PUT OUT ENTIRE LINE
01450 NEXT     LD     A,0DH        ;PUT NEWLINE TO LISTING
01460          CALL   WRITE
01470          LD     A,(3840H)    ;CHECK KEYBOARD
01480          CP     4            ;FOR BREAK KEY
01490          JR     NZ,LINE      ;IF NOT BREAK, NEXT LINE
01500 ;
01510 END      LD     A,1AH        ;PUT END-OF-FILE FLAG TO TAPE
01520          CALL   WRITE
01530          CALL   CASOFF       ;CASSETTE OFF
01540          CALL   KWAIT        ;WAIT FOR USER
01550          JP     START
01560 ;
01570 ASCII    LD     IX,DECTBL    ;CONVERT BINARY INTEGER IN
01580                              ;HL TO CHARACTER FORM
01590 ASCII1   XOR    A            ;ZERO COUNTER
01600          LD     B,(IX+1)     ;GET POWER OF 10 IN BC
01610          LD     C,(IX)
01620          OR     A            ;CLEAR FLAGS
01630 ASCII2   SBC    HL,BC        ;DIVIDE BY REPEATED SUBTRACTION
01640          JR     C,ASCII3     ;UNTIL VALUE GOES NEGATIVE
01650          INC    A            ;STILL POSITIVE, INCREMENT COUNT
01660          JR     ASCII2       ;AND TRY AGAIN
01670 ASCII3   ADD    HL,BC        ;ADD BACK
01680          ADD    A,0B0H       ;TO ASCII WITH BIT 0 SET
01690          CALL   WRITE        ;PUT OUT RESULT
01700          LD     A,C          ;ARE WE FINISHED
01710          CP     1
01720          RET    Z            ;YES, LAST DIVISION WAS BY 1
01730          INC    IX           ;NO, GET NEXT POWER OF TEN
01740          INC    IX
01750          JR     ASCII1
01760 DECTBL   DEFW   10000        ;VARIOUS POWERS OF TEN
01770          DEFW   1000         ;FOR DIVISION
01780          DEFW   100
01790          DEFW   10
01800          DEFW   1
01810 ;
01820 WRITE    PUSH   DE           ;PUT OUT BYTE VIA SUITABLE ROUTINE
01830          PUSH   HL
01840          DEFB   0CDH         ;'CALL' OP-CODE
01850 WRITER   DEFW   0000H        ;FILLED IN WITH APPROPRIATE
01860                              ;ROUTINE ADDRESS
01870          POP    HL
01880          POP    DE
01890          RET
01900 ;
01910 LISTR    AND    7FH          ;KILL BIT 0
01920          CP     0DH          ;NEWLINE?
01930          JR     Z,NEWLIN     ;YES, SPECIAL ACTION
01940          CP     09H          ;TAB CHARACTER
01950          JR     Z,TAB        ;YES, SIMULATE A TAB
01960          CALL   OUTPUT       ;PUT OUT A BYTE
01970          LD     HL,CURSOR    ;KEEP CURSOR IN STEP
01980          INC    (HL)
01990          RET
02000 ;
02010 NEWLIN   PUSH   DE
02020          CALL   OUTPUT       ;PUT OUT NEWLINE CHARACTER
02030          POP    DE
02040          CALL   ADDRES       ;ADDRESS OF NEXT LINE
02050          LD     A,5          ;CURSOR POS TO 5
02060          LD     (CURSOR),A
02070          CALL   DELAY        ;SLOW LISTING TO READABLE SPEED
02080          LD     A,(HALT)     ;SEE IF WE WANT TO PAUSE
02090          OR     A
02100          JR     Z,CHECK
02110          CALL   GETKEY       ;PAUSE UNTIL KEYBOARD INPUT
02120          CP     1            ;BREAK ?
02130          JP     Z,START      ;THEN ABANDON LISTING
02140          CP     0DH          ;NEWLINE/ENTER ?
02150          RET    NZ           ;NO, CARRY ON
02160          XOR    A            ;YES, CANCEL HALT MODE
02170          LD     (HALT),A
02180          RET
02190 ;
02200 CHECK    LD     A,(3840H)    ;SEE IF ANY CONTROL KEY PRESSED
02210          OR     A
02220          RET    Z
02230          LD     A,1          ;YES, SET HALT MODE ON
02240          LD     (HALT),A
02250          RET
02260 ;
02270 TAB      PUSH   BC           ;SIMULATE A TAB
02280          LD     A,(CURSOR)   ;GET CURRENT POSITION
02290          SUB    11           ;ALLOW FOR ADDRESS AND LINE NUMBER
02300 TABLP1   SUB    10           ;DIVIDE CURSOR POSN BY 10
02310          OR     A            ;EXACTLY DIVISIBLE BY 10 ?
02320          JR     Z,TAB10
02330          JP     P,TABLP1     ;JUMP IF STILL POSITIVE
02340          NEG                 ;NOW NEGATIVE, NEGATE VALUE
02350          LD     B,A          ;AND SKIP THAT MANY SPACES
02360          JR     TABLP
02370 TAB10    LD     B,10
02380 TABLP    LD     A,32
02390          CALL   OUTPUT
02400          LD     HL,CURSOR    ;KEEPING CURSOR IN STEP
02410          INC    (HL)
02420          DJNZ   TABLP
02430          POP    BC
02440          RET
02450 ;
02460 OUTPUT   DEFB   0CDH         ;OUTPUT A BYTE
02470 OUTR     DEFW   0000H        ;TO APPROPRIATE ROUTINE
02480          RET
02490 ;
02500 ; DELAY ROUTINE TO SLOW LISTING TO READABLE SPEED
02510 ;
02520 DELAY    LD     BC,(WAIT)    ;LOAD DELAY DURATION
02530          CALL   PAUSE        ;AND CALL DELAY ROUTINE
02540          RET
02550 ;
02560 ; CONVERT BINARY INTEGER IN DE TO ASCII HEX REPRESENTAION
02570 ;        AND OUTPUT
02580 ;
02590 ADDRES   PUSH   HL           ;PUT OUT ADDRESS IN DE
02600          PUSH   BC
02610          PUSH   DE
02620          LD     A,D          ;GET MSB
02630          CALL   HEX          ;CONVERT TO ASCII AND OUTPUT
02640          POP    DE           ;RESTORE ORIGINAL INPUT
02650          PUSH   DE           ;AND SAVE IT AGAIN
02660          LD     A,E          ;GET LSB AND PRINT
02670          CALL   HEX
02680          LD     A,32         ;PRINT SPACE
02690          CALL   OUTPUT
02700          POP    DE
02710          POP    BC
02720          POP    HL
02730          RET
```

```
02740 ;
02750 HEX    LD    C,A       ;CONVERT BYTE IN A TO ASCII HEX
02760        SRL   A         ; AND OUTPUT RESULT
02770        SRL   A         ;ISOLATE MSB BY SHIFTING
02780        SRL   A         ;LSB OFF TO RIGHT
02790        SRL   A
02800        CALL  HEX1      ;CONVERT TO ASCII AND PUT OUT
02810        LD    H,A
02820        LD    A,C       ;RESTORE INPUT ARGUMENT
02830        AND   0FH       ;ISOLATE LSB
02840        CALL  HEX1
02850        LD    L,A
02860        RET
02870 ;
02880 HEX1   PUSH  BC        ;CONVERT BYTE IN A TO ASCII
02890        ADD   A,30H
02900        CP    3AH       ;0 - 9
02910        JP    M,HEX2    ;A - F
02920        ADD   A,7       ;A - F
02930 HEX2   CALL  OUTPUT    ;PUT OUT RESULT
02940        POP   BC
02950        RET
02960 ;
02970 GETNAM EQU   .
02980        LD    HL,NMSG   ;ASK FOR FILE NAME
02990        CALL  OUT
03000        LD    A,32
03010        LD    HL,NAME
03020        LD    B,6
03030 BLANK  LD    (HL),A    ;BLANK OUT NAME BUFFER
03040        INC   HL
03050        DJNZ  BLANK
03060        LD    HL,NAME
03070        LD    B,6
03080 FILL   PUSH  BC
03090        CALL  GETKEY    ;GET CHARACTER
03100        CP    8         ;BACKSPACE ?
03110        JR    Z,GBKSP   ;SAVE IT
03120        PUSH  AF
03130        PUSH  HL
03140        CALL  VIDEO     ;AND ECHO ON DISPLAY
03150        POP   HL
03160        POP   AF        ;RESTORE REGISTERS
03170        POP   BC
03180        CP    0DH       ;ENTER/NEWLINE
03190        RET   Z
03200        LD    (HL),A    ;ACCEPT CHARACTER IN BUFFER
03210        INC   HL
03220        DJNZ  FILL
03230        RET
03240 GBKSP  POP   BC        ;RESTORE COUNTER
03250        LD    A,B
03260        CP    6
03270        JR    Z,FILL    ;CANT GO BACK
03280        PUSH  BC
03290        LD    A,8
03300        PUSH  HL
03310        CALL  VIDEO     ;BACKSPACE AND ERASE DISPLAY
03320        POP   HL
03330        POP   BC
03340        INC   B         ;ALLOW ANOTHER CHAR
03350        DEC   HL        ;ADJUST POINTER
03360        JR    FILL
03370 ;
03380 GETHEX LD    HL,0      ;CONVERT INPUT BUFFER CONTENTS
03390                        ;TO A BINARY VALUE IN HL
03400        LD    IY,INBUF  ;IY POINTS TO INPUT BUFFER
03410 GHEX1  LD    A,(IY)    ;GET VALUE FROM BUFFER
03420        INC   IY        ;BUMP PTR
03430        CP    0DH       ;IF NEWLINE, THEN FINISHED
03440        RET   Z
03450        CP    3AH       ;IF 0 - 9
03460        JR    C,GHEX2
03470        SUB   7         ;FOR A - F
03480 GHEX2  AND   0FH       ;ZAP BITS 0 - 3
03490        ADD   HL,HL     ;MULTIPLY PREVIOUS RESULT BY 16
03500        ADD   HL,HL
03510        ADD   HL,HL
03520        ADD   HL,HL
03530        LD    D,0       ;AND ADD IN NEW HEX DIGIT
03540        LD    E,A
03550        ADD   HL,DE
03560        JR    GHEX1
03570 ;
03580 ;    MEMORY EDIT FACILITY
03590 ;
03600 EDIT   CALL  CLS       ;CLEAR SCREEN
03610        LD    A,14      ;TURN ON CURSOR
03620        CALL  VIDEO
03630        LD    HL,VIDEO
03640        LD    (OUTR),HL ;PLANT VIDEO ROUTINE
03650        LD    HL,AMSG   ;EDIT MEMORY
03660        CALL  OUT
03670        CALL  INPUT     ;GET INPUT
03680        LD    A,(INBUF) ;GET FIRST CHAR OF BUFFER
03690        CP    13        ;ENTER/NEWLINE = NO INPUT
03700        JR    Z,EDIT    ;YES, TRY AGAIN
03710        CALL  GETHEX    ;CONVERT INPUT TO BINARY IN HL
03720 ELOOP  CALL  SHOHL     ;OUTPUT WORKING MEMORY ADDRESS
03730        LD    A,200
03740        CALL  VIDEO     ;PRINT 8 SPACES
03750        PUSH  HL
03760        LD    A,(HL)    ;GET BYTE FROM MEMORY
03770        PUSH  AF        ;SAVE IT
03780        CP    32        ;CAN WE DISPLAY IT AS ASCII ?
03790        JR    C,NODISP  ;ELIMINATE CONTROL CODES
03800        CP    128       ;AND GRAPHICS
03810        JR    C,DISP
03820 NODISP LD    A,'-'     ;CAN'T DISPLAY, USE '-'
03830 DISP   EQU   .         ;DISPLAY RESULT
03840        CALL  VIDEO
03850        LD    A,200     ;SKIP 8 SPACES
03860        CALL  VIDEO
03870        POP   AF        ;RESTORE MEMORY BYTE
03880        CALL  SHOWA     ;CONVERT TO ASCII AND DISPLAY
03890        LD    A,200     ;SKIP 8 MORE SPACES
03900        CALL  VIDEO
03910        CALL  INPUT     ;GET USER INPUT
03920        LD    A,(INBUF) ;FIRST INPUT CHAR
03930        CP    13        ;ENTER/NEWLINE = NO INPUT
03940        JR    Z,GOON    ;YES, NO MEMORY CHANGE
03950        CALL  GETHEX    ;NO, NEW VALUE IS IN INPUT BUFFER
03960                        ;CONVERT IT TO BINARY IN HL
03970        LD    A,H       ;H VALUE SHOULD BE ZERO
03980        OR    A         ;OTHERWISE, ERROR

03990        JR    NZ,ERROR
04000        LD    A,L       ;INPUT VALUE TO 'A' REGISTER
04010        POP   HL        ;RESTORE MEMORY POINTER
04020        LD    (HL),A    ;LOAD MEMORY WITH NEW VALUE
04030        PUSH  HL
04040 GOON   LD    A,0DH     ;PUT OUT A NEWLINE
04050        CALL  VIDEO
04060        POP   HL
04070        INC   HL        ;STEP TO NEXT MEMORY LOCATION
04080        JR    ELOOP
04090 ERROR  LD    A,'?'     ;INPUT BYTE ERROR
04100        CALL  VIDEO     ;ASK FOR REPEAT INPUT
04110        POP   HL        ;RESTORE CURRENT MEMORY POINT
04120        JR    ELOOP     ;AND GO BACK
04130 ;
04140 SHOHL  PUSH  HL        ;SHOW ADDRESS IN HL
04150        EX    DE,HL
04160        CALL  ADDRES
04170        POP   HL
04180        RET
04190 ;
04200 SHOWA  PUSH  HL        ;CONVERT VALUE IN 'A' REGISTER
04210                        ;TO ASCII, AND OUTPUT
04220        CALL  HEX
04230        POP   HL
04240        RET
04250 ;
04260 ; ROUTINE FOR INPUT OF HEX VALUES
04270 ;
04280 INPUT  LD    HL,INBUF  ;GET INPUT
04290        LD    B,0       ;COUNTER
04300        LD    C,5
04310 INP1   PUSH  BC
04320        CALL  GETKEY    ;LOOK AT KEYBOARD
04330        POP   BC
04340        CP    10        ;DOWN ARROW ?
04350        JR    Z,ENDINP  ;YES, ABANDON INPUT
04360        CP    0DH       ;NEWLINE/ENTER ?
04370        JR    Z,ENDINP  ;YES, INPUT COMPLETE
04380        CP    8         ;BACKSPACE ?
04390        JR    Z,BKSP    ;YES, TAKE ACTION
04400        CP    30H
04410        JR    C,INP1    ;LT 0 ?
04420        CP    3AH       ;)9 ?
04430        JR    C,ACCEPT
04440        CP    'A'       ;TEST A - F
04450        JR    C,INP1
04460        CP    'G'
04470        JR    NC,INP1
04480 ACCEPT LD    (HL),A    ;PARK IN BUFFER
04490        PUSH  HL
04500        PUSH  BC
04510        CALL  VIDEO     ;DISPLAY ON VIDEO
04520        POP   BC
04530        POP   HL
04540        INC   HL        ;BUMP POINTER
04550        INC   B         ;AND COUNTER
04560        LD    A,B       ;CHECK COUNT
04570        CP    C         ;CHECK FOR BUFFER FULL
04580        JR    Z,ENDINP
04590        JR    INP1      ;GET ANOTHER INPUT BYTE
04600 BKSP   XOR   A         ;CAN'T BACKSPACE PAST BUFFER START
04610        CP    B
04620        JR    Z,INP1
04630        DEC   HL        ;OK, DECREMENT POINTER
04640        DEC   B         ;AND ADJUST COUNTER
04650        LD    A,8       ;PUT OUT BACKSPACE/DELETE CHAR
04660        CALL  VIDEO     ;SEND BACKSPACE TO VIDEO
04670        JR    INP1
04680 ENDINP LD    A,0DH     ;PUT NEWLINE CHAR IN BUFFER
04690        LD    (HL),A
04700        CALL  VIDEO     ;DISPLAY NEWLINE
04710        RET             ;AND EXIT
04720 ;
04730 GETKEY PUSH  HL
04740        CALL  KWAIT     ;GET KEYBOARD INPUT IN 'A'
04750        CP    1         ;IS IT BREAK ?
04760        JP    Z,START   ;YES, RESTART
04770        POP   HL
04780        RET
04790 ;
04800 ; ****************************************************
04810 ;
04820 MSG    DEFM  'EDTASM BUFFER RESCUE'
04830        DEFB  0DH
04840        DEFB  0DH
04850        DEFM  '  Q  -  Quit, Reboot System'
04860        DEFB  0DH
04870        DEFB  0DH
04880        DEFM  '  L  -  List Buffer on VDU'
04890        DEFB  0DH
04900        DEFB  0DH
04910        DEFM  '  P  -  List Buffer on Printer'
04920        DEFB  0DH
04930        DEFB  0DH
04940        DEFM  '  W  -  Write EDTASM Compatible Source Tape'
04950        DEFB  0DH
04960        DEFB  0DH
04970        DEFM  '  E  -  Edit Memory'
04980        DEFB  0DH
04990        DEFB  0DH
05000        DEFB  0
05010 READYC DEFB  0DH
05020        DEFM  'READY CHOICE :'
05030        DEFB  0DH
05040        DEFB  0
05050 .
05060 NMSG   DEFB  0DH
05070        DEFM  '....'
05080        DEFB  0
05090 AMSG   DEFB  0DH
05100        DEFM  'GIVE ADDRESS'
05110        DEFB  0
05120 ;
05130 NAME   DEFM  'REMOLE?'
05140 CURSOR DEFB  0
05150 HALT   DEFB  0
05160 WAIT   DEFW  0000H
05170 INBUF  DEFS  5
05180 ;
05190 BUF    DEFW  ????
05200 ;
05210 ;
05220        END   START
```

in ASCII format, except that the leftmost bit is set in each byte. Thus, zero is represented as B0 Hex, rather than ASCII 30 Hex. Line number 100 is therefore represented on tape as B0 B1 B0 B0.
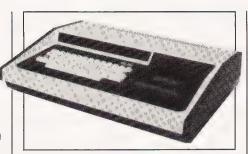
Byte 5 is a space (20 Hex).

The text of the line then follows, with Tab represented as 09 Hex, and the line is terminated by a carriage return character, 0D Hex.

The end of file is indicated by a flag byte with value 1A Hex.

USING THE PROGRAM

After completing the development, I reloaded my source buffer system tape dump, loaded the rescue program, and tried to list the buffer. I obtained a perfect listing from lines 100 to 380, but line 390 was incompletely listed. After line 390, the cursor performed a series of gymnastics, and gibberish appeared on the screen, but from line 520, the listing resumed normally to the end. Inspection of line 390 using the memory examine/edit facility showed that my rogue program had somehow converted the text length counter to 03, and by counting the bytes in the line, I realised that the correct

value was 1B. I edited the offending memory location, created a EDTASM-compatible source tape via the tape write facility, loaded EDTASM-PLUS, loaded my newly created tape, and was back in business. Since then, I have used the program on several occasions and estimate that I have now fully recovered the development effort, plus at least a man-week in normal programming time.

Sometimes, one is not so lucky as in the example given above, and large chunks of the source code may have been made unrecognisable by a runaway test. In these cases, by experimenting with dummy text counts until a reasonable listing can be obtained it is usually possible to produce a partial recovery tape which can be laoded into EDTASM and edited to restore the missing portion. When

all else fails, it is always possible to cut one's losses by forcing termination of the buffer by inserting 0FFFF Hex as the line number after the last recognisable source line, and recover up to that point. At least the whole thing is not lost as was the case before introduction of the rescue program.

PROGRAM NOTES

At any point, when entering input, one can return to the main menu via the Break key. The Backspace/Left-arrow key is honoured for correction of errors. When examining memory, the Down-arrow or the Enter/Newline key can be used to step through memory leaving the values unchanged.

The listing options include the ability to run a continuous list, or step through line by line. Pressing the space bar sets the listing in single-step mode, and the Newline/Enter key sets continuous listing mode. The VDU listing starts normally in single-step mode, and the printer listing starts in continuous mode. For the VDU listing, a short delay between lines is included to permit the listing to be read as it scrolls past.

The unbeatable TI Home Computer. It's all the computers your family will ever need.

Buying a home computer is something you have to get right first time. It's too late when you've got one to find it won't take plug-in software. Or can't be programmed without an expensive accessory.

The TI Home Computer is a real computer system

The TI Home Computer has got the memory power you might expect from more expensive computers, built in. At its heart is a powerful TMS 9900 16-BIT Microprocessor. Most other home computers have only an 8-BIT. And you can expand the memory from 16K of RAM up to 52K.

The total memory capacity is 114K Bytes.

A wide range of software for everyone

Another feature that makes the TI system so powerful, yet so easy to use is Solid State Software.™ These plug-in cartridges cover everything from space games like Parsec™ to teaching maths, managing home finances and composing music. And the range is getting wider all the time.

It even has what professionals look for in a home computer

CPU: TMS 9900 16-BIT, plus 256-byte Scratchpad RAM.
Memory: Total 114K bytes; 26K bytes ROM internal; up to 36K ROM cartridges external; 16K built-in RAM expandable to 52K bytes.
Keyboard: 48 Key QWERTY, alpha lock, function key auto repeat.
Sound: 5 octaves, 3 simultaneous tones, noise tone.
Colour: 16 foreground and background. High resolution.
Interfaces: Cassette, TV, 2 joysticks, main peripheral port.

™ trademark of Texas Instruments.

More than one programming language

The standard programming language, TI BASIC, is built into your TI Home Computer so you can begin programming right away. But there's an expanded range of optional languages like Extended BASIC, TI Logo, USCD-Pascal, TI FORTH and Assembler.

With these you can fully expand your programming skills.

A wide range of peripherals

Most computers lose a lot of memory when you add peripherals. The TI Home Computer is different. Every peripheral comes with its own built-in programs to keep the loss of memory to a minimum.

The convenient Peripheral Expansion System houses up to eight peripherals. Additional hardware cards simply plug in. You can even add a complete Floppy Disk Memory System.

The peripherals include memory expansion, RS232 Interface, P-Code card and more. There's also a sophisticated matrix printer and Solid State Speech™ synthesizer – which you can use with your own TI BASIC programs.

A lot more for no more

The TI Home Computer gives you so much more without costing more. At today's price it's exceptional value. Take your family round to try one. If you never try it you'll never know what you're missing.

TEXAS INSTRUMENTS
Creating useful products and services for you.

Peter Green

A LOOK AT THE LASER

Hailing from the sky-scrapered shores of Hong Kong, the Laser 200 is a surprisingly late arrival from this Land of Technology. Has it been worth the wait?

What is this I see before me? Looking rather like a well-fed, albino version of the ZX Spectrum, the Laser 200 is a rather late entry into the low-cost home computer market from Hong Kong. Quite typically, this origin means that it's very cheap indeed — the basic unit retails for £70. However, there is rather more to the story than simply a low price tag, so let's dig a little deeper and see how appealing the Laser is.

A CASE IN POINT

The Laser has been designed along the same general lines as the ZX Spectrum. Covering a slightly larger area than the Spectrum and about twice as thick, it consists of little more than a sloping keyboard with the electronics tucked in underneath. The keys are made of the same hard rubber (or dead flesh, depending on your point of view) as the Spectrum, and number 45 rather than the latter's 40. The case is cream with a dark brown keyboard surround and light brown keys — all the key legends are in white and are easy to read. An LED at the top right of the keys indicates when the computer is powered up.

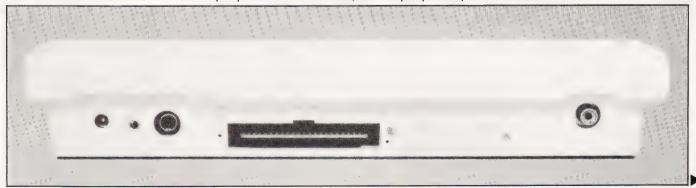Like the Spectrum, the Laser 200 allows single keystroke entry of BASIC keywords: but unlike the Spectrum it doesn't insist on them. This is good. Beginners will be able to spell the words out in full to begin with, gradually changing over to single key entry as they learn where all the various functions are located. This is easier to pick up than on the Spectrum, where the keyword locations are sometimes a little illogical: on the Laser, words that form natural groupings are located on adjacent keys (like IF-THEN-ELSE, FOR-TO-STEP-NEXT, SET-RESET-POINT and PEEK and POKE). Furthermore all the words in a given grouping need the same type of Shift operation to get the keyword.

There are two function keys, Shift and Control, and none of the other keys has more than four functions. Unshifted, the keys produce the alphanumeric set and some of the punctuation. Pressing Shift with a key gives the rest of the punctuation, the arithmetic operators and the block graphics. Control and a key gives the BASIC keyword marked above the key, while Control-Return, then Control-key gives the keyword below. (This latter procedure is similar to Sinclair's extended mode). One oddity when using single-key entry; if the keyword requires brackets, as in STR$ (X), then for some functions the leading bracket is printed for you, sometimes it isn't. Oh, well, just remember to keep your eyes on the screen.

On our way round to the back of the computer, we take a slight detour on to the right-hand side where an on/off switch is located. There is, strictly speaking, no real need for this as the Laser isn't mains-powered but uses a separate low voltage power pack like most other computers of this size. However, it's marginally more convenient to flip the switch for a hard reset, should you need one, then reach round and pull out the plug. A trifling point, really.

From left to right across the back panel we have, first of all, the 9 V DC input socket for the power supply, then the tape socket. Yes, socket — singular. Unusually, the Laser has a stereo jack socket rather than the normal twin sockets, but it does have a tape lead supplied with the required connector and the standard

The back of the Laser 200. The memory expansion bus is visible, but the peripheral port is shuttered.

The 16K RAM expansion plugged in. This lies flat rather than sticking up like Sinclair's version.

plugs at the cassette end. No remote control of the cassette recorder motor is provided.

Next comes a monitor output, rare (and commendable) in a machine of this price, followed by the two printed circuit board edge connectors for the memory expansion and peripherals. Finally comes the UHF TV output socket, tuned to Channel 36 or thereabouts as usual.
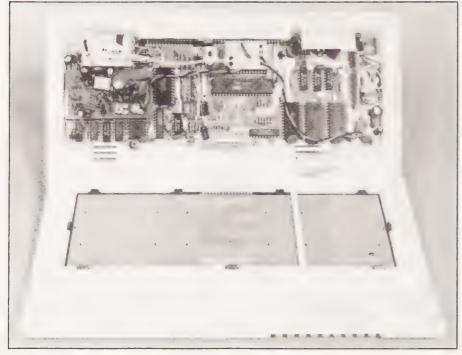
Thus endeth the guided tour. Also included in the purchase price are a TV lead (too short for comfortable viewing with a domestic TV set, like most other computers), a User manual, a demonstration tape, a BASIC Reference manual and a slim booklet of example programs.

TURNING ON

On power-up the Laser 200 simply says READY. No Microsoft copyright message (for it is they who wrote the BASIC), no message giving the number of bytes free — just READY and a flashing cursor. You can't check how much free memory there is with FRE(0) or SIZE, since such a statement isn't supported. The display is yellow text on a green background, which I promptly messed up by POKEing random graphics all over the place to see what the screen capabilities were.

This led to an interesting discovery when I tried to clear the screen. There is no key provided for clearing the screen, so it's necessary to use CLS in immediate mode. But with random patterns on the screen the remainder of the line must be cleared with spaces to prevent a syntax error. In doing this, I overshot onto the next line and instead of overwriting that too, the Laser 'opened up' a new line by scrolling the remainder of the screen down a line. An attempt to repeat this on the next line failed, as the cursor refused to move past the end of the second line. The point of all this is that the BASIC is designed to prevent the input of anything longer than two lines, and since the screen is only 32 columns wide, program lines can only be 64 characters long including the line number. This is rather less than the 80-character lines Microsoft normally allows.

Another annoying feature is the action of the Delete key. Instead of being a combined backspace-and-delete, it is necessary to use the cursor keys to position the cursor over the first of the offending characters. Delete then removes that character and pulls the end of the line back by one character, so making a correction could take twice as many keypresses as usual. Fortunately the auto key repeat speeds things up but it was a little difficult to get used to.
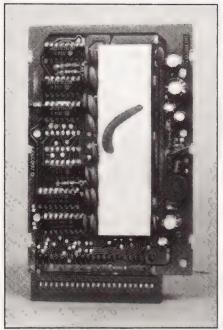
Apart from these quirks the BASIC is pretty much standard Microsoft, with multistatement lines, the usual maths functions, the usual string handling functions (sufficient memory for string operations must be reserved using CLEAR), and the surprising IF-THEN-ELSE which some more expensive machines do not have. Arrays can have up to three dimensions. I/O functions are supported by INP and OUT, and USR calls to machine code routines may be made.

The cassette commands are the standard CLOAD, CSAVE and VERIFY, plus CRUN which loads a program and autoruns it. For some reason the manual insists in quite strong terms that you must always start the tape running before hitting Return during any tape operation: I can understand this for CSAVE, where you might lose some of the header, but not for the other three, and the machine didn't complain when I broke the rules. Named data files may be stored on tape using PRINT #, and loaded into variables using INPUT #.

GRAPHICS

There are always two graphics modes. The text mode, which the Laser always defaults to when a program isn't running, is MODE(0) — it

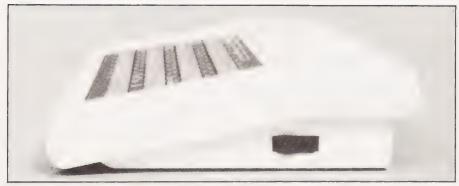It isn't terribly tidy inside the Laser, but everything seems to work OK.

insists on the brackets – and gives a 32 by 16 display. Text is upper case only, with a choice of yellow on green or light brown on dark. Selecting inverse text gives you the same two colour combinations with the foreground and background reversed. Since the Laser uses a separate display code for each of the normal and inverse characters, that takes care of half of the possible 256 displayable characters. The other 128 display codes are assigned to eight repetitions of the 16 text mode block graphics characters, one set for each of the eight foreground colours in this mode (green, yellow, blue, red, buff, cyan, magenta and orange). You can have any background colour you like for the block graphics so long as it's black. Indeed, the only way to get black on the screen at all is as part of a text mode graphics block.

Inside the RAM pack we find the memory chips under metal shielding, and a switched mode psu down the right of the PCB.

Note the use of display codes rather than ASCII codes: like the PET, Sharp and other machines, to get an 'A' on the screen you can either PRINT CHR$(65) or POKE 28762,1.

In the high-res graphics mode, MODE (1), the pixel resolution is 128 by 64, rather poor by today's standards. The colour set is also restricted in this mode, with a choice of two sets. There's a green background with green, yellow, blue and red foreground colours, or a buff background with buff, cyan, magenta and orange foreground. No text can be displayed in MODE (1), and the only pixel manipulation commands are limited to SET, RESET and POINT (returns the colour of the

An on-off switch is provided on the side of the Laser.

tested pixel). No line drawing commands, no CIRCLE, no flashing from hardware. Sigh.

Resorting to machine code can give much better possibilities, as in the 'intro' and 'outro' sections of the demo tape. This program is not recommended for epileptics!

SOUND

The SOUND command is not much of an improvement on that of the Spectrum, though it is louder. Two parameters can be specified, to give 31 frequencies and nine different durations. OK for simple tunes and games sound effects, but nothing advanced.

EXPANSION

The 4K user RAM of the basic Laser 200 may be expanded by the addition of a 16K module, which we tested, or a 64K module, which we didn't. The module seemed rather chunky compared to RAM packs for other computers and we couldn't resist opening it up to take a look. Underneath the layers of metal, presumably for RF shielding, we discovered a small switched mode power supply, amongst other things. This is probably generating 12 V and suggests that the price has been kept down by using the older multi-rail supply chips, rather than the modern single rail 5 V versions.

The peripheral port will take an add-on printer interface which will drive the Seikosha GP-100 and GP-100A printers (according to the manual), or any Centronics printer (according to the synopsis on the

packaging). The relevant commands are LLIST, LPRINT and COPY; the manual doesn't go into details about what happens to the various colours when the high-res screen is dumped.

Again, according to the packaging there is a light pen and a joystick which may be connected to the peripheral port, though no mention is made of how to program for them. The details are probably included with the accessories, and we were not supplied with either.

The question of possible disc drives is even more vague: the only reference to them is in the list of error messages at the back of the manual, which includes DISK COMMAND as one entry.
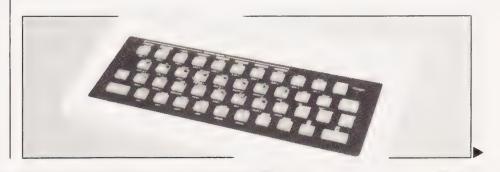
THE DOCUMENTATION

The manuals for the Laser 200 are no worse than those for many other computers, and are better than some. There's the usual smattering of spelling mistakes, most of which are harmless, and the level is pitched at the rank beginner. Unfortunately the manual has equated simplicity with brevity in many places, and a beginner may need rather more explanation of some aspects. The sample programs are all short and quite basic.

I particularly liked page 21, which had a drawing of the screen with SYNTAX ERROR displayed on it, and beneath it the explanation, "This means SYNTAX ERROR   "

CONCLUSIONS

Throughout his review I have made comparisons between the Laser 200 and the Spectrum, which one tends to

do instinctively given their similar appearance. In reality this is probably an unfair comparison, because although the Laser costs only £70, the basic computer has only 4K of memory and the price of a 16K RAM pack takes the price up to that of a 16K Spectrum, which offers much better graphics and more facilities for expansion now that the Microdrives and networking are available. (On the other hand, the 64K expansion takes the price to that of a 48K Spectrum). Perhaps a fairer comparison for the basic machine would be one made with the ZX81, another computer intended as a low-cost entry into computing but with an inferior keyboard and no sound and colour.

Unfortunately Sir Clive, with his usual consummate timing in these matters, has just dropped his price to £45 for a ZX81, 16K RAM pack and software cassette, forcing people to decide whether it is worth paying the extra £25 for sound, colour and a quarter of the memory: not to mention the vast amount of software available for the two Sinclair machines which widens the gap even further. It seems that the Laser 200 has fallen between several stools, and it may remain there unless the distributors can stimulate the interest of the commercial software houses.

BENCHMARK TIME	BM1	BM2	BM3	BM4	BM5	BM6	BM7	BM8	Average
	1.7	7.0	17.0	17.4	19.3	31.6	48.8	72.5	26.9

FACTSHEET

CPU	Z80A
ROM	16K
RAM	4K (expandable to 16 or 64K)
Language	Microsoft BASIC
Keyboard	45-key multifunction, moving rubber membrane
Display	Text mode: 16 lines of 32 characters, 32 by 64 pixel graphics in eight colours plus black. High-res mode: 64 by 128 pixels in four colours (choice of two sets), no text. TV or monitor output
Cassette	600 baud
I/O	Centronics printer interface, lightpen, joysticks, bus for memory expansion
Sound	Single channel, 31 notes, 9 durations
Costs	Laser 200 £69.95
	16K RAMpack £29.95
	64K RAMpack £59.95
	Printer interface £19.95
	Joysticks £19.95 per pair
	Lightpen £19.95
Supplier	Computers For All, Southfields Industrial Park, 30 Hornsby Square, Laindon, Essex Telephone 0268 418414

sinclair special

4

ZX Microdrive

ZX Microdrive

1

Inside...
Two special offers...
Six new software titles...
Microdrive!

ZX Microdrive System preview!

ZX MICRODRIVE
At least 85K bytes storage, loads a typical 48K program in as little as 9 seconds: £49.95.

ZX MICRODRIVE CARTRIDGE
Compact, erasable, revolutionary. Complete with its own storage sleeve. Contains up to 50 files, with a typical access time of 3.5 seconds: £4.95.

ZX INTERFACE 1
Necessary for sending and receiving information from ZX Microdrive. Includes RS232 interface and local area network facility for 2 to 64 Spectrums. Attaches to the underside of your Spectrum. Purchased with ZX Microdrive, just £29.95. As separate item, £49.95.

PSYCHOLOGY, GRAND PRIX RACING, BRAIN TEASING, PROGRAMMING, SPACE-BLASTING!

Sinclair have it all taped with six brand-new programs for ZX Computers!

Chequered Flag
For 48K RAM Spectrum. **£6.95**

Have you ever wanted to drive a Formula One car flat-out round a Grand Prix circuit? With Chequered Flag you'll need one eye on the road and one eye on the instruments, as you steer and brake to avoid hazards, and work through the gears in search of the lap or race record. This outstanding new program puts you in the driver's seat with stunning realism, and gives you a choice of three cars and ten different circuits. Don't crash!

Mothership
For ZX81 with 16K RAM. **£4.95**

Scream down the claustrophobic confines of the Zarway. Engage suicidal drone fighters in deadly laser combat. Dodge, duck and dive in a high-speed 3-D race to attack the evil Mothership before she claims your home planet. Mothership is a truly tough challenge, and fast, furious fun!

Cattell IQ Test
For 48K RAM Spectrum. **£12.95**

Although there are a number of so-called self-administered IQ tests on the market, the only reliable way of finding your IQ has – until now – been to visit a qualified psychologist and take a battery of tests – for a fee. Now Victor Serebriakoff, International President of Mensa, has produced Professor Cattell's test in a form which enables you to use your ZX Spectrum to test your IQ.

The Cattell Scale IIIA test is timed by the computer, marked immediately, and the marks standardised against your age. This is the first time that an accredited, standardised test has been available to the general public.

Zeus Assembler
For 48K RAM Spectrum. **£12.95**

A powerful and easy-to-use programming aid, designed to simplify the entire process of producing machine code programs, enabling you to write in assembly language instructions. Comes complete with comprehensive range of support facilities.
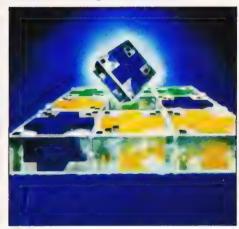
Monitor and Disassembler
For 16K & 48K RAM Spectrums. **£12.95**

This powerful Disassembler translates machine code into comprehensible assembly language instructions, allowing you to examine the BASIC ROM, to investigate the workings of the Spectrum or to analyse your own machine code routines.

With the highly versatile Monitor, you get an extensive set of facilities to aid the entry, inspection, modification and debugging of your own machine code programs.

Flippit
For 16K or 48K RAM Spectrums. **£9.95**

Like those cube games, Flippit looks simple. But its fiendish ingenuity results in the ultimate game of logic and patience. Twist, turn and swap the nine Flippit pieces in search of the elusive magic square. But be warned, those pieces can be arranged in *millions* of combinations…

Garry Marshall

BOOK PAGE

Our reviewer discusses a handful of books of an educational nature, good for children of all ages.

The books reviewed this month are primarily about computer languages and learning. Two of them are designed for learning a computer language at school. Happily, they manage to convey more than just learning a language by showing what computers can be used for and, to some extent, how they should be used. This relates to the crux of the argument of those who object to BASIC as a programming language which, as everybody knows by now, I suppose, is that while it may be easy to learn BASIC it is not easy to learn to write good programs in BASIC. Another two books are about how computers can, and should, be used to educate. This emphasis on education does not mean that the books are only for use in school or college. A good learning environment is just as important for those of us who are not students whether we are at home or at work. The final book is about Information Technology. This is a subject which, in view of the way things are moving, we should all know something about and, anyway, all uses of computers fall within its sphere.

Beginning COMAL is written by Borge Christensen, the Dane who invented the language COMAL. This language is a hybrid of Pascal and BASIC which retains the friendly operating environment of BASIC and adds to the small repertoire of features possessed by BASIC the simple structured features of Pascal. In this way it is claimed that COMAL combines the best of both BASIC and Pascal. However that may be, COMAL has been adopted by the Education Ministries of several European countries (including Denmark, naturally enough) as the official programming language for use in their schools.

The book is essentially a programmed learning text that guides those with no previous knowledge of computing through to a fairly complete knowledge of COMAL and how to use it. It is clearly based on material used and developed in the classroom: this has the advantage that the material is tried and tested. There are resulting drawbacks, though. The book is a collection of teaching material designed to be used while sitting at the computer. Unfortunately, the material has not been particularly well edited, and the material is not organised into a coherent work. A particularly glaring example of this is that as early as the fourth page of Chapter 1 we are told to LOAD "EXE17". Obviously, the students for whom the material was developed had a suite of programs available that could be loaded into their computers, but, as the readers of the book, we have not. A little thought and quite a lot of flicking through the pages of the book reveals that there is a section giving program listings towards the back of the book and program EXE17 is listed under the heading 'Exercise 1.7'. Nowhere in the book is this explained.

The features of COMAL are introduced throughout the book by means of interesting programming examples, and a top-down style of program development is used consistently. Assignments are introduced to provide the values required to fill the gaps in a standard letter, decision making is the basis of an arithmetic drill program, file handling is introduced to show how stock control is done in a small shop, and so on.
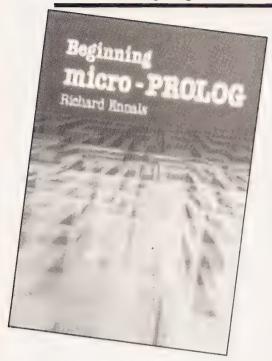
All in all, the book provides a good introduction to COMAL as well as to developing programs in COMAL and the uses to which COMAL programs can be put. This can all be learnt quite painlessly, but if interest should flag there is always the odd spot of fourth form humour or 'Danish English' to bring some relief. It is a pity that the index is so sparse since this limits the book's value as a source of reference for the language.

Beginning micro-PROLOG is in many ways a very similar book to the one on COMAL. Its major chapter representing the core of the book, is a direct presentation of the teaching material used to present micro-PROLOG in the classroom. The language itself is a version of PROLOG developed by the Logic Programming group at Imperial College, London, to which the author Richard Ennals belongs. It is designed to be run on microcomputers and to be used by children. PROLOG is the language associated with the development of the fifth generation computers.

A very gentle introduction to micro-PROLOG using many examples and exercises is presented. Because the teaching material is intended for use with quite young children, most older readers will find it very easy going. The temptation to skim over pages of simple examples can cause an important point in the development to be missed, however. A 33 page section gives the complete answers to even the most trivial exercises.

The main chapter is surrounded by five more rather short ones ▶

These deal with important matters such as why we should use logic as a programming language, what part logic plays in the school curriculum, and how PROLOG can be used for other things, including Expert Systems. One can only regret that these chapters were not more fully written as they cover matters that provide the justification and the motivation for the use of PROLOG.

The justification that is given for the use of PROLOG is worth reciting. It starts with the observation that the first question most people ask about any program is 'What does it do?'. The answer they expect is something like 'It reads a collection of records and sorts them into alphabetical order'. That is, they expect to be told the relation between the input and the output of the program. They do not expect as an answer a listing of the program. Similarly, the most effective way to specify a programming task is to give a specification of what the program must do to its inputs to produce the required output. Surely, then, a natural way to program a computer is to give it a description of the task it should perform in the form of the relation between the input and output. PROLOG is a descriptive language which permits programming to be done in this natural way.

In contrast, languages such as BASIC and Pascal (and COMAL) are imperative languages with which programs are written as a series of commands. As people do not think purely in terms of commands, these languages are less compatible with the way that people think than is PROLOG. This in turn relates to the fifth generation

computer project, one of the aims of which is to make computers more accessible for people.

Anyway, **'Beginning micro-PROLOG'** provides a painless introduction to PROLOG as well as showing how it can be used. It also conveys a good deal about data bases and something about Expert Systems. Although it is quite good as it is, the expenditure of a little more time and effort could have turned it into a really excellent introduction to many matters of current interest.

Mindstorms by Seymour Papert is already a famous book. The blurb on its cover says that the book is 'all about LOGO — how it was invented and how it works'. That seems to me to be misleading. As I read the book it is about how computers can be, and are being, used as tools to help us to learn and to expand our minds generally. The book doesn't aim to teach LOGO, but it does show how it can be used and what it can be used for. In fact, if you try to find out from the book what LOGO is, expecting the answer that it is a computer language, you will be surprised to discover that Papert regards it as a philosophy of education. LOGO attempts, in much the same way as PROLOG does, to release the potential of computers for us in ways that conventional languages cannot.

One of Papert's fundamental ideas is that the way for us to use computers, particularly in education, is for us to program the computer and not for it to program us. This sounds so obvious that it is hardly worth repeating, but if you consider how much software there is that lets you use the computer **exactly** as you would like to, the force of the idea may begin to strike you. Certainly in schools at present, drill programs are much more prevalent than programs that allow any innovative form of learning. The same difficulty also affects many people who, having learnt the elements of BASIC, have the greatest difficulty (and not surprisingly) in seeing how it can help them to use the computer for their complex business or personal requirements.

The most widely known aspect of LOGO is the Turtle, which can be controlled from LOGO with commands such as FORWARD, BACK, LEFT and RIGHT. The Turtle is a prime example of what Papert calls a 'microworld' in which a restricted world is established by the computer in which exploration and discovery can take place. The creation of a 'microworld' to which children, in particular, can relate provides tremendous opportunities

for learning, testing ideas and developing mental patterns and structures, all of which are fundamental to mental development.

In case this give the impression that microworlds are only relevant to children, consider that a Flight Simulator is a microworld. It is not usually considered as such, and it has nothing to do with Papert and LOGO, but that is essentially what it is. If a computer can provide the environment in which such a complex skill as flying a jumbo jet can be learnt completely, there is surely no limit to what can be learnt in the appropriate microworld. Since one of Papert's claims for LOGO is that it helps us to learn about learning, and given that we shall all have to learn new skills and ideas during our working life in an increasingly technical age, the relevance of the ideas in **Mindstorms** is readily apparent.

Another important idea is that of a 'bug' in a program. This makes it possible to get away from the common idea many people have that something is either right or wrong, and that there are no shades in between. Since getting something wrong carries the connotation of failure it is naturally discouraging. Turtle programs, just like any others, seldom work first time. But rather than allowing this to mean that the program is 'wrong', Papert encourages the attitude of wondering whether a program can be fixed. Thinking in terms of whether a program can be fixed, rather than whether it is right or wrong, is the prelude to the skill of correcting bugs. This attitude is essential for a programmer, but it is also valuable in many other spheres of life.

Papert's book contains much more than I have been able to indicate here. It will not all be absorb-

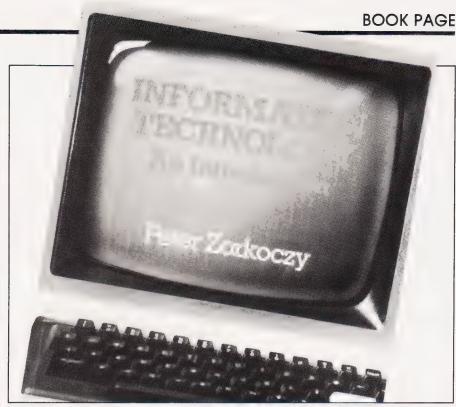ed in one reading. But this book must be read by anyone interested in using computers to their full potential.

Turtle Geometry by Harold Abelson and Andrea diSessa is a book that builds on ideas presented in Papert's book by using the Turtle as a basis for exploring geometry. If you think that geometry is a dry mathematical subject you won't recognise it as treated in this book. If you like geometry this book will make you like it even more.

The approach to Turtle geometry differs from that of ordinary geometry in some very fundamental ways. In ordinary geometry a circle is represented by a formula such as $x^2 + y^2 = r^2$. In Turtle geometry a circle is represented by a program telling a

Turtle how to follow a circular path. Thus, Turtle geometry deals with programs rather than equations, and is a computational form of geometry. (Naturally, the programs are written in LOGO.) Another difference is that in ordinary geometry a formula describes a complete curve, such as a circle, giving what is called a 'global' description. In Turtle geometry a curve is described a little at a time, and the Turtle only knows at any time what its present move should be. Although when given a program for a circle, the Turtle will inevitably trace a circle, it has no way of knowing this, since it only knows what its current small movement is. This kind of description is called 'local'.

The opening chapter of the book deals with how to tell a Turtle to trace some simple closed paths or, if you like, how to draw simple closed

curves, introducing the necessary aspects of LOGO in the process. The second chapter covers a wide range of interesting topics, including how a Turtle can be programmed to simulate an animal finding prey using its sense of smell, how two Turtles interact when one is the predator and the other the prey, and how complex designs can be produced with simple recursive programs. Chapter 4 deals with the topology of Turtle paths and culminates with a program to allow the Turtle to escape from any maze. The program incorporates the Pledge algorithm devised by John Pledge at the age of 12. Subsequent chapters deal with the Turtle on curved surfaces and the book finishes in the realms of general relativity!

Incidentally, how do you think the Turtle can follow a straight path when it is moving on a sphere or some other curved surface? The answer is that the Turtle is going straight when the step taken by its left legs has the same length as the step taken by its right legs. (If one step were shorter than the other the Turtle would be turning and not going straight.)

Turtle geometry is, as the blurb correctly states in this case, about using the computer as a means for exploring mathematics, although it is much more interesting than this makes it sound. The book is an endless source of fascination and stimulation.

Finally, **Information Technology — An Introduction** by Peter Zorkoczy is an attempt to examine what is meant by Information Technology (IT). Although last year was IT year, there still seems no real agreement about what is meant by

IT. This book should help to establish a consensus. In fact, Richard Ennals gives a rather better short definition of IT than anything that Zorkoczy attempts. According to Ennals, IT is 'the handling of vocal, pictorial, textual and numerical information by means of a microelectronics-based combination of computing and telecommunications'. By including chapters on telecommunications, computers, voice communication with computers, computer vision and electronic mail the book implicitly endorses this definition.

The first part of the book consists of two chapters on the fundamentals and the applications of IT. The second part has 13 short chapters, each giving a short technical description of IT in a particular application. Disappointingly, the chapters are almost independent, painting a picture of IT as a lot of isolated applications rather than as having any single, unified existence. However, the book provides the best introduction to IT that is currently available to my knowledge.

The books reviewed this month are:

Beginning COMAL by Borge Christensen (1982) published by Ellis Horwood, 332 pages, £6.90.
Beginning micro-PROLOG by Richard Ennals (1983), published by Ellis Horwood, 192 pages, £6.50.
Mindstorms by Seymour Papert (1980), published by The Harvester Press, 230 pages, £4.50.
Turtle Geometry by Harold Abelson and Andr a diSessa (1981), published by the MIT Press, 469 pages, £14.00.
Information Technology — An Introduction by Peter Zorkoczy (1982), published by Pitman, 134 pages, £4.50.

● This is an entirely new computer system. It was designed with both eyes fixed firmly on the future. So that whatever shape the future takes, we'll be able to fit it into the system. Just as simply as the peripherals and software already available fit into the system. That way, the system will grow with you. And you'll never get left behind.

● Based around the Z80A microprocessor, and utilising Microsoft™ BASIC, Aquarius™ has 8K ROM and 4K RAM resident within its console. It is able to provide up to 16 colours and resolution of 320 x 192, and generates its sound directly through the television's speakers.

● With twin cartridge ports, the mini-expander allows simultaneous use of additional RAM and software cartridges. Twin disc game hand controls are included and the unit provides two additional sound channels. The 16K RAM cartridge plugs into either the console or the mini-expander, increasing Aquarius™'s RAM capacity to 20K.

● With the ability to reproduce the entire graphic and character set of Aquarius™ at 80 characters a second, the printer's 40 column output allows transcription of the complete monitor image.
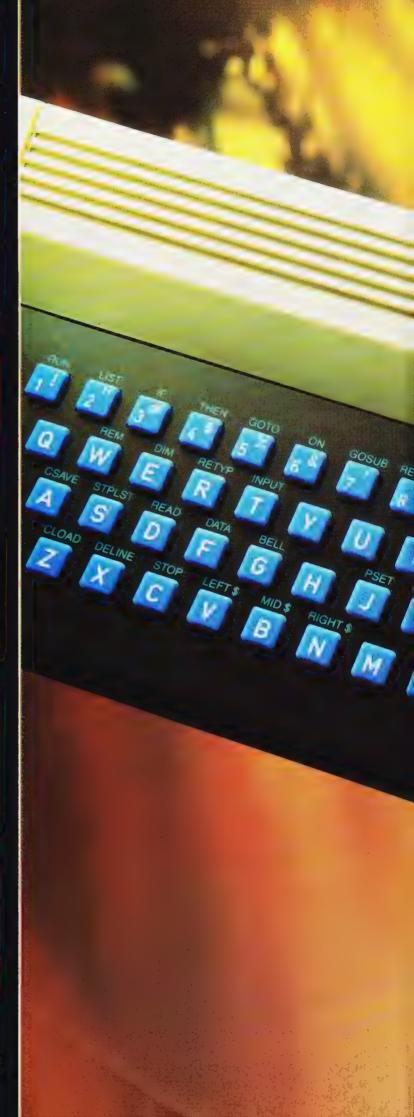
● Using standard audio cassettes, the data recorder provides storage for programs and information, and allows the use of cassette based software. Incorporating a digital tape counter and transmission indicator, it operates sequential searching.

● A large number of games, designed to take advantage of Aquarius™'s sophisticated colour and sound capabilities, are available on cartridges that plug into the console either direct, or through the mini-expander. Cassette based games can be used via the data-recorder.

● A wide range of preprogrammed cartridges is available, including the LOGO teaching program and practical home data systems like FILEFORM™ and the spreadsheet calculator package, FINFORM.™

AQUARIUS™
HOME COMPUTER SYSTEM

WITH AQUARIUS, YOU WON'T GET LEFT BEHIND.

MATTEL ELECTRONICS ®

ADVENTURES

BEING AN ACCOUNT OF THOSE COMPUTER PROGRAMS COMPRISING QUESTS, RIDDLES, MONSTERS AND MACHINES, AND THE PROGRAMMING THEREOF

SOLVING GAMES

There are two sorts of Adventure games — one with many variations within the game, the other with a fixed, single path to the final objective. The first type is a variation of the role-playing game known as Dungeons and Dragons (a registered trademark of TSR Hobbies Inc), in which one or several players wander around a scenario collecting treasure, experience, bruises, and sometimes a trip to the afterlife. In this type of Adventure luck plays a large part and so a winning strategy cannot be given here. CT's The Valley was such a game.

This supplement is more concerned with the second type of Adventure, which is now becoming immensely popular. In this version, there is a single objective which can usually only be obtained in one way. The point of the game is to find this single solution, which involves solving what amounts to a series of riddles or logical problems posed by the author of the game. The answer to one riddle will reveal more of the game and new riddles, so everything has to be solved in the right order — if a problem has you stumped, you can't just skip it!

The first games of this type to be implemented on home computers were written by Scott Adams, of whom almost all of you will have heard. Versions of his games exist for most common microcomputers. Other series include the Mysterious Adventure games, the Acornsoft Adventures such as Castle of Riddles and Countdown to Doom, the text/graphics adventures from Phipps Associates, and the famous Hobbit from Melbourne House. This latter adventure is possibly the current state-of-the-art game, although it is a little peculiar in that various characters other than the player seem to be wandering around doing their own thing.

On commencing the game, the player generally finds himself in a starting location, with or without possessions, either with a goal of finding treasure, overcoming a dire fate, or sometimes both. You wander around from location to location, finding objects, picking some up and moving others from room to room, trying to solve the various puzzles using the objects at your disposal either singly or in conjunction with one another. There may be secret doors, magic words, spaceships, weapons, mazes, monsters, and anything else the programmer can dream up provided that it's consistent in the context of the plot, and fair on any player possessing reasonable intelligence and cunning.

MOVING ABOUT

One of the main functions of the description of each location is to tell you in which of the possible locations you can move. However, sometimes the description may not mention every possible exit. This may be due either to sloppiness on the part of the programmer, or maybe because he wants to make sure you're you're paying attention! Just because a direction isn't mentioned, it doesn't mean that an exit that way does not exist. So a golden rule is — always try moving in every possible direction in every location (unless it's explicitly barred). This can be very time consuming, especially in a program written in BASIC rather than machine code, but it's better than finding out after days of fretting that an 'impassable' room had an exit after all! Also remember to try out different action verbs if it seems appropriate: it may be impossible to GO NORTH because of a cliff, but it may be possible to CLIMB CLIFF.

This approach has saved me a long delay on at least two commercial Adventures. In one, the description of a certain location suggested that beyond the jungle was a huge glacier, yet after finding my way through the jungle there was no glacier to be seen. In fact you got to the glacier by going UP at the original location, but this was not obvious from the description.

On a second occasion I found myself on a cliff ledge with a description saying "There is a slit in the ground near your feet". That doesn't really suggest that you could fit through it, but typing "DOWN" got me into a subterranean cavern.

MAPPING

As you explore the game, of course, you'll have to make a map of the places you've visited, the ways in which they interconnect and what lies in each. Most people have devised their own system for this — there's no hard-and-fast rule so long as your map is legible and easy to follow. For each location you need to include a brief and unique title (MAP ROOM, CELLAR, MOUNTAIN SLOPE and so on), a note of any objects you find there, and paths to adjacent scenes.

For my own system I draw square boxes with the room title in capital letters at the top, plus brief notes describing the location and its objects underneath. You should note down *any* nouns used in the description which might be important, even if you seem to be unable to interact with them just now. Later on you may discover an artifact which makes a use for these objects obvious. Incidentally, since you don't know the position of your starting location relative to the rest of the map, you might as well start in the middle of the paper (which should be quite large!) unless the exits all seem to be in one general direction, in which case you can start in the appropriate corner of the paper.

Lines to other boxes are drawn in to indicate pathways. Give yourself room between boxes or things can get a bit cluttered, especially in Adventures where all eight points of the compass are valid directions. Up and down movements complicate things still further, and you may need to break a map up into sections on separate sheets of paper if too many rooms cluster together. This is OK provided you make it quite clear how the various sections are linked together.

Up and down can be indicated by curving lines 'on top of' and 'beneath' the boxes. Routes which only allow travel in one direction can be marked with an arrow to show that they're one-way. Any special requirement for using a given route can be jotted down next to the relevant lines (for example, need magic carpet and wand).
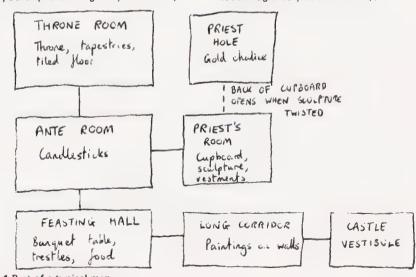
Fig. 1 Part of a typical map.

MAZES

Something that you tend to come across in almost every commercial Adventure game sooner or later is a maze — whether the game needs it or not! A lot of authors treat it simply as an easy way of making the player do some hard work, rather than stretch their own imagination to devise some new and more challenging puzzles. There are only a couple of variations on the classic maze theme and it really seems pointless to keep including these when the methods of mapping them are so well-known.

A maze in an Adventure game is a series of locations all with the same description, typically something like "You are in a room with exits in all directions". The rooms are interconnected in a complex pattern (in fact, sometimes you pass through an exit and end up in the same room, while other times leaving the way you came in doesn't lead you back to the room you just left). This means that simply wandering about aimlessly is going to get you nowhere fast — particularly so when mazes generally have only one exit. But how can you tell where you are when all the places look alike? How do you map a maze?

Well, like Hansel and Gretel you have to leave a trail of objects behind you as you go. Unlike Hansel and Gretel, each object must be different in nature. The procedure is to drop an object — any object — in the first room and leave it there. Draw a box on your map to represent this room, note down what you left there, pick any exit and move onto the next location.

One of two things will happen. Either you'll come back to the room you were just in (you'll know because of the object you left there) or you'll be in a different, empty room. If you're in an empty room, drop a second object and mark a second room on your map. In either case, indicate against the exit you took from the first room which location it lead to. Now pick another exit and move on. Again, if you come to an empty room, drop another object to 'label' it and note it against the previous exit. Now you can just carry on exploring the maze, dropping objects in each empty room and eliminating all the interconnecting passages one by one until the whole maze is mapped.

There are, of course, two points to note about this procedure. The first is that you don't know in advance how many rooms the maze contains, so you'll need to be carrying as many objects as the game permits (most Adventures have an upper limit on your inventory). Running out of objects halfway through a maze can be a very frustrating experience! A devious programmer might make the number of rooms in a maze equal to the maximum number of items that you can carry. A
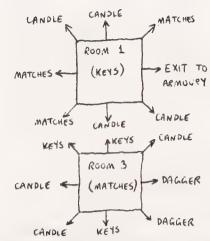
Fig. 2 Mapping a maze. Don't draw in all the links!
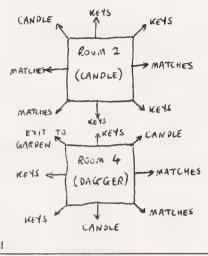
very devious programmer could make it equal to the number of items plus one, so you'd run out of items with one empty room left. This would still be a uniquely identifiable location, of course, but it might well cause worries until the game player realised there was only *one* empty room left. Alternatively, some programmers put more rooms than the items in the inventory, but leave one or more items lying around in the maze for you to find.

The second point to remember is that, if you should die during your exploration of the maze, when you attempt to run it the second or subsequent times to complete the map, *you must start off by following your route on the first trip through, dropping the same items in the same rooms.* Otherwise your additions to the map won't correspond to the same rooms in the first attempt and you'll get hopelessly lost. Remember, directions in a maze are illogical by definition.

Another type of maze is one in which there are subtle differences in the description of each maze location — differences which the programmer hopes you won't notice! Go through the description letter by letter, count the number of commas and full stops very carefully ... Of course the mapping procedure is the same as before, but you don't have to keep dropping objects, since each location is uniquely identified, albeit subtly.

Keep the map simple, as shown here, by labelling the exits with their destinations. If you try to draw lines between all the boxes to indicate all the possible routes you'll end up with a tangled mess that's nearly impossible to interpret. Once you've accounted for every pathway and identified the exit(s), then you can run the maze through on your subsequent attempts at the game without dropping objects. Just make sure you don't forget which room you're in when you get halfway through the maze!

A final word on mazes — in one game a maze was featured into which

you were not permitted to take any objects at all. Once you were inside you found that some of the rooms contained objects, but some didn't, and passage through was apparently reduced to the sort of trial and error approach that is exactly what we've been trying to avoid. If you come across a maze like this, all I can say is — you're on your own!

PROBLEM SOLVING

Now you can explore the Adventure, map it intelligently and clearly, and cope with any mazes that you might come across. That just leaves the puzzles, and there it is much more difficult to offer helpful advice. After all, they are the whole point of the game and they're different for each one (unless you've bought a program from a plagiarist!).

Some general thoughts may be of help, however. Don't ignore *anything* as irrelevant: in a good Adventure, anything you come across will probably turn out to have a function. If you haven't found anything in the game that could help with a particular obstacle, can you *adapt* something? For example, if you've got a bottle but no knife and you need to cut some rope, try breaking the bottle and using the sharp glass. Try combining two or more of the objects at your disposal. Think laterally — if you need to undo a screw and there aren't any tools to be found in the game, will the point of a dagger double as a screwdriver? Examine *everything*. If the game features graphics then scrutinize the picture for clues: if it has sound then listen to the tunes, write down the titles and if necessary the lyrics too. Clues may be hidden *anywhere*.

Think of old sayings, proverbs, literary references, nursery rhymes even; if you find a lamp, the odds are that rubbing it will accomplish something! You might even gain an insight into a possible solution by considering the personal 'style' of an author whose previous games you've solved. And if all else fails, remember a lot of other people are probably in the same boat.

Maybe if you could find an oar somewhere ...

WRITING YOUR OWN ADVENTURES

Although this supplement aims to explain how you can write your own text Adventure games, it's really only possible to tell half the story. While the programming details can be covered here, the success or otherwise of an Adventure depends a great deal on the imagination behind the plot, the skill with which the traps are sprung and the conundrums posed, and there isn't anything we can do to teach you those qualities. But we can give a few pointers to good plot technique.

Over the page you'll find a diagram of a sample Adventure we've designed especially for this supplement. It's been written to illustrate many of the features typical of this sort of game, and we'll start off by examining the plot.

In this scenario you are the pilot of a star cruiser who, through some unspecified whim of Fate, has managed to crash his ship quite spectacularly into some cliffs on an uncharted planet. The only thing to survive the crash besides yourself is your spacesuit. If you try to leave the ship without putting your suit on first, you are poisoned by the atmosphere. (The reason why the plot requires the wearing of the suit to be compulsory is revealed later). Outside the ship, which has burst apart on impact, you find a lead-tungsten alloy shielding plate from the engine room. Further south there is a mutant star goat grazing on the pasture land, which resists all your attempts to move it.

Going back the other way you go through a mountain pass, and if you climb the cliff to the north you find a lump of uranium which has broken off of a weathered outcrop of the ore. Two other pieces of uranium can be found in similar locations leading off the pasture, but if you try to pick up and carry all three pieces at once they attain critical mass and melt down, killing you in a burst of heat and radiation. The correct procedure is to take the lead shield to the west end of the ice cave and leave it there, and place two of the uranium nuggets on the floor at the east end of the cave. Then go back to get the third nugget and throw it from behind the safety of the lead shield. The lead protects you and the nuggets melt a hole down through the ice floor to reveal a tunnel. If you try to climb down the tunnel, you slip and break your neck.

The lead shield has another function, however. The mercury lake has a deep space radar at the bottom, which you need to retrieve. But you cannot help but float like a cork on the mercury. Stepping into the lake while carrying the shielding allows you to sink to the bottom and get the radar, but now the only way back up to the surface is to drop the shielding, which means you've lost it for good. Therefore you must melt the ice cave floor *before* going after the radar, otherwise you cannot complete the game.

Taking the route through the mountain pass, leads to an ironwood thicket, which bars your access to the north bank of the stream. Further east the thicket expands into a full-fledged forest, within which you find a tree that can be climbed. Up in its branches you discover the skeleton of a rather careless traveller who has accidentally flown into the tree and killed himself. The skeleton is still wearing the lift belt, and by taking it and wearing it you have a means of descending from the ice cave to the tunnel.

There is also a woodman's hut in the forest, which contains a machete. Aha, you think, with this I can cut down the thicket — and when you try it the blade shatters against the ironwood. That's not what the machete is for! Going round past the goat again and to the east you come to a boat tethered on the south bank of the stream. The boat is actually a red herring, it's the rope that you need. However, you also need the bottle of Thranx beer which is in the boat, so the various possibilities are as follows. You can't untie the tight knot with your gloved hands (see why you're forced to wear the spacesuit?), so you have to cut it with the machete. If you cut it before getting the beer, the boat is swept away and you lose it. If you cut the rope while you're standing in the boat,

you're swept over the waterfall and killed. And if you try to take a short cut to the north bank by swimming across the stream, you still get swept away.

Using the rope you can tie up the goat and lead it round to the ironwood thicket, which it promptly devours. (I did say it was a mutant!). This allows you to reach the north bank of the stream, where someone has dropped a key. You can now descend to the rocky tunnel beneath the ice cave and go out onto the plateau, where you find that the key will gain you entance to the Tardis. However, the mercury has run out of the fluid link and it needs refilling, but the lift belt hasn't got enough power left to get you back up. What you should have done was pour away the beer (a waste, but you can't drink beer while you're wearing a spacesuit), and fill the empty bottle with mercury from the lake before leaving this part of the game.

With the Tardis repaired you materialise on a new planet, a desert world. If you wander off in any direction you die, but fortunately the Tardis appeared under a working spaceship, which you steal. (This is an Adventure, not a morality play). However, you immediately fly into a thick dust nebula which makes navigation impossible, because the deep space radar on this ship is dead. This constitutes the 'classic maze' of our game, and provided you brought the salvaged deep space radar from the mercury lake with you, you will be able to distinguish the different locations and map it properly.

However, your ship will run out of fuel before the end of the game unless you land on the planet in location 39, whereupon automatic refuelling of the ship's tanks takes place. It is also necessary to land on the small moon in location 45, leave the ship and collect the aerial from the wreckage on the abandoned landing pad (locations 47, 48). Finally, once you find your way out of the nebula you can land on the last planet, repair the distress beacon and send off an SOS to your rescuers.

Notice how the physical layout of the Adventure has been designed to mislead the player and trick him into falling for booby traps which hinder the solutions of the puzzles proper. For example, by arranging things so that the player comes across the ironwood thicket and *then* the machete, we plant the thought that he can cut down the shrubs, whereas the machete has an entirely different purpose. Similarly the lead shield has two functions, the first in the ice cave and the second in the lake, but by presenting them to player in the opposite order you can increase the chances that he'll get things wrong. A game in which you find some keys and a locked door in the next room, or a torch and a dark location two steps away, are not very satisfying

to play.

This kind of misdirection is perfectly legal: after all, magicians provide their entertainment by fooling you into seeing what they want you to see, and your main aim as game designer is to entertain the player. He doesn't necessarily have to like you as well!

While you are allowed to be vicious, you are required to be fair. Be consistent in your plotting. Don't have magic amulets and dragons turning up in a basically science fiction Adventure, and don't leave laser cannons lying around in ruined Mayan temples. This confuses the issue and looks very sloppy. And remember that being fair means sticking to things which are reasonable and accurate in the context of the scenario.

An example of what I mean came up while I was plotting the Adventure overleaf. I originally made the engine shielding out of lead: after all, it's radiation-proof and it's heavy. Then I explained the plot to a colleague, and he pointed out that mercury is denser than lead and so you wouldn't sink into the lake! Back to the periodic table of the elements, where tungsten turns out to be denser than mercury. I wasn't sure whether tungsten made a good radiation shield, though, so a lead-tungsten alloy for engine room shielding was suddenly invented! It's this sort of attention to detail that sorts out the good Adventures from the merely average, and you should strive for it at all times.

STARTING OUT

Once you've got the plot for your Adventure sorted out in your mind, you can make a map like the one overleaf for it. At this stage there's no real need to go into great detail and flowery prose about all the location descriptions. That doesn't become important from the point of view of the actual programming until much later.

What you must ensure at this stage is that the logical structure of the

Adventure is correct. It becomes very messy later if you want to start altering the number of rooms or the connections between them.

The method of mapping used in our sample Adventure is one that works quite well: it is similar to that described in the previous section on solving commercial Adventures. Each location is represented by a box, lines between boxes indicate connecting routes, and an arrow on a route indicates that it's one way only. Connections between locations that are made by means of teleport, or that don't appear until certain conditions have been met, can be represented on the map by dotted lines.

Give the bare minimum of information in each box — its title, and what objects are found within it. Each box should also have a unique number, starting from 1. We use location 0 for a special purpose, as you'll see later.

Some Adventure designers advocate planning your Adventure on paper marked up with squares or octagons, with adjacent locations touching. This is fine provided you're writing an Adventure where all the locations are adjacent and equidistant, but often this isn't the case. Look at the Tardis, for example. In our game there's only the control room to consider, but a different game could have many rooms in the Tardis to be explored. Yet the nature of a Tardis is that it's bigger on the inside than on the outside, so from the outside it should appear the size of a single location only. The boxes and lines method makes this sort of situation easier to represent on paper.

You can also note down natural obstacles to movement around the outside of the boxes with this system, to serve as a reminder when you finally start composing the finished place descriptions that will appear in the final game. Words like cliffs, canyon, dense forest will do at this stage.
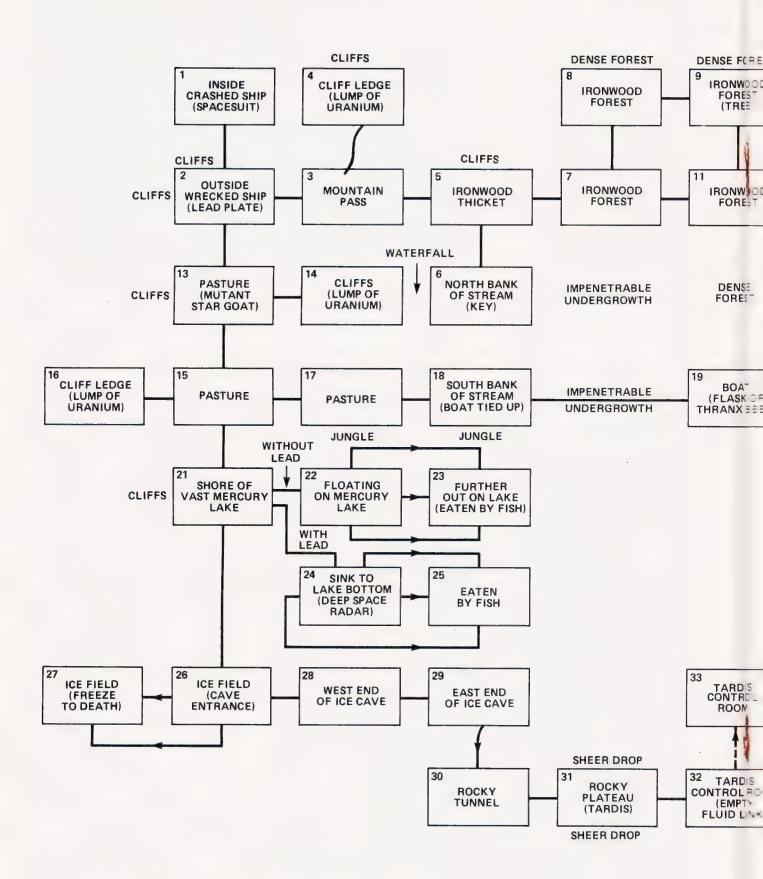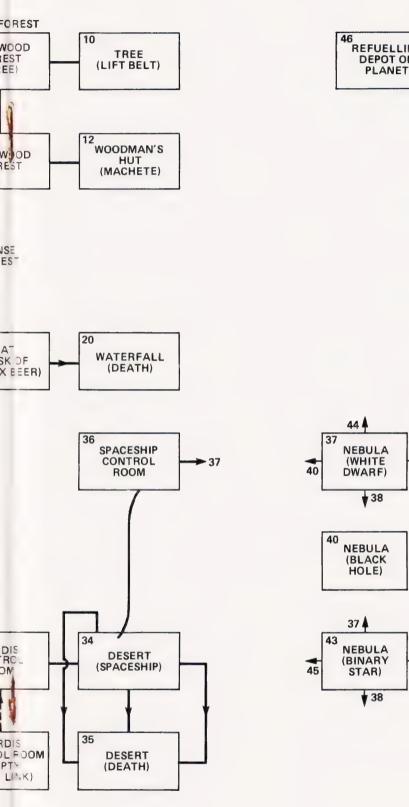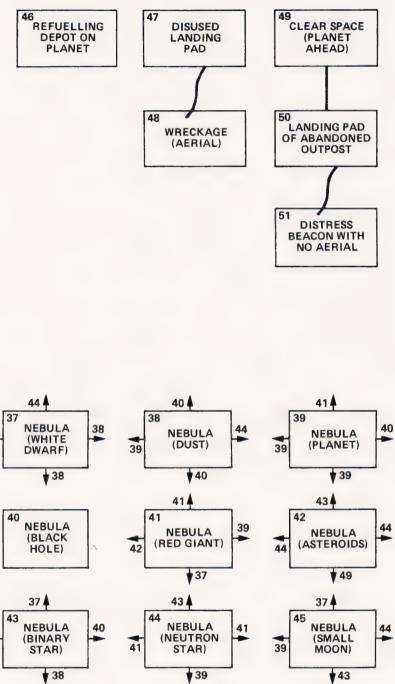
Fig. 3 A sample Adventure map. Only simple notes need to be made at this stage: the more complicated descriptions of each location can be made later (see the next page).

FOREST

WOOD
REST
EE)

| 10 | TREE (LIFT BELT) |

WOOD
REST

| 12 | WOODMAN'S HUT (MACHETE) |

NSE
REST

| 46 | REFUELLING DEPOT ON PLANET |

| 47 | DISUSED LANDING PAD |

| 49 | CLEAR SPACE (PLANET AHEAD) |

| 48 | WRECKAGE (AERIAL) |

| 50 | LANDING PAD OF ABANDONED OUTPOST |

| 51 | DISTRESS BEACON WITH NO AERIAL |

A
SK OF
X BEER)

| 20 | WATERFALL (DEATH) |

| 36 | SPACESHIP CONTROL ROOM | → 37

DIS
RO
OM

| 34 | DESERT (SPACESHIP) |

RDIS
L ROOM
PT
LINK)

| 35 | DESERT (DEATH) |

44 ↑
| 37 | NEBULA (WHITE DWARF) | → 38
40 ←
↓ 38

40 ↑
| 38 | NEBULA (DUST) | → 44
39 ←
↓ 40

41 ↑
| 39 | NEBULA (PLANET) | → 40
39 ←
↓ 39

| 40 | NEBULA (BLACK HOLE) |

41 ↑
| 41 | NEBULA (RED GIANT) | → 39
← 42
↓ 37

43 ↑
| 42 | NEBULA (ASTEROIDS) | → 44
← 44
↓ 49

37 ↑
| 43 | NEBULA (BINARY STAR) | → 40
45 ←
↓ 38

43 ↑
| 44 | NEBULA (NEUTRON STAR) | → 41
41 ←
↓ 39

37 ↑
| 45 | NEBULA (SMALL MOON) | → 44
39 ←
↓ 43

1. INSIDE CRASHED SHIP.
You are inside the control cabin of your spaceship. Everything is smashed, but the exit port to the south is still functioning. Your spacesuit is lying on the floor.

2. OUTSIDE WRECKED SHIP.
You are standing outside what is left of your once sleek and powerful star cruiser. Now it is scrap. Cliffs block the way north and west. There is a path to the east, and a vast pasture lies south. A lead-tungsten shielding plate has been blown from the engine room and is lying on the ground.

3. MOUNTAIN PASS.
You are on a narrow path winding through a mountain pass. The cliffs to the south are sheer, those to the north rough and jagged. The path leads east into some vegetation.

4. CLIFF LEDGE.
You are standing on a narrow cliff ledge high above the pass. There is a metallic outcrop here which has been badly weathered. A lump of uranium is lying on the ground.

5. IRONWOOD THICKET.
You are on a path which leads between sheer cliffs to the north and a thicket of ironwood shrubs to the south. The path continues east.

6. NORTH BANK OF STREAM.
You are standing on the north bank of a wide stream. To the west there is a vast waterfall tumbling down from the cliffs. To the east the ironwood shrubs are impenetrable. Someone has dropped a key on the bank.

7. IRONWOOD FOREST.
You are on a path leading through an ironwood forest. The path continues east, and the forest appears less dense to the north.

8. IRONWOOD FOREST.
You are standing in an ironwood forest. The forest appears less dense to the east and the south.

9. IRONWOOD FOREST.
You are standing beneath a huge tree in an ironwood forest. The forest appears less dense to the west and the south.

10. TREE.
You are clinging to the branches in the upper part of an ironwood tree. There is the skeleton of a careless Flier lodged in the branches. The skeleton is wearing a lift belt.

11. IRONWOOD FOREST.
You are standing in an ironwood forest on a path that runs east-west. To the east lies the door of a woodman's hut.

12. WOODMAN'S HUT.
You are in a deserted woodman's hut. There's some old furniture lying around and no-one seems to have been here for a long while. There is a machete hanging on the wall.

13. PASTURE.
You are standing on a vast plain of pasture land. Mountains bar the way west, while a gentler slope leads up into the cliffs lying east. The pasture extends to the south. A mutant star goat is quietly grazing here.

14. CLIFFS.
You are standing at the top of a waterfall which plunges from the cliffs down to a stream lying east. There are several weathered outcroppings of metal ore here. A small lump of uranium is lying here.

15. PASTURE.
You are standing on the southern end of the pasture land. What could be a climbable path winds into the cliffs to the west, while the pasture extends to the north and the east. There is a vast shiny area visible in the distance to the south.

16. CLIFF LEDGE.
You are on a cliff ledge above the pasture land. Weathered outcroppings of metal are visible here and there, and a winding path leads east down to the pasture. There is a small lump of uranium lying here.

17. PASTURE.
You are standing on the eastern part of the pasture. The pasture continues to the west, while cliffs lie north and thick jungle lies south. The route east seems clear.

18. SOUTH BANK OF STREAM.
You are standing on the south bank of a wide stream flowing east from a huge waterfall. The jungle grows down to the water's edge making further passage east impossible. There is a boat tied up to a tree here.

19. BOAT.
You are standing in a small boat tethered to a tree on the south bank of the stream. There is a flask of Thranx beer lying in the bottom of the boat.

20. WATERFALL.
The boat is caught helplessly in the current and you are powerless to stop it. Another huge waterfall appears to the east, and the boat plunges over it.

21. SHORE.
You are on the shore of a vast lake which seems to consist of pure mercury. The lake stretches away to the east as far as the eye can see. Cliffs block the way west, while to the south it appears to get even brighter.

22. LAKE.
Mercury is pretty dense stuff, you know, and you're floating on the surface like a cork.

23. LAKE.
You're floating further out from the shore where the mercury is deeper. There's a sudden disturbance beneath the surface, and as you disappear down its gullet you realise you have discovered a new species of fish.

24. LAKE BOTTOM.
The lead-tungsten shield you are carrying is much denser than the mercury, and you sink rapidly to the bottom of the lake. As you hit bottom you feel your feet touch some object.

25. LAKE BOTTOM.
As you wander further out into the deeper mercury, there is a sudden disturbance and a rather large predator fish devours you whole.

26. ICE FIELD.
You are standing on an immense ice field which stretches to the horizon to the west and south. To the north you can see the lesser glitter of the mercury lake, while a cave entrance is visible in the ice peaks to the east.

27. ICE FIELD.
As you wander further into the icy region, your suit heater gives up under the strain. You promptly freeze to death.

28. WEST END OF CAVE.
You're standing in the western end of a huge ice cave. The cave exit lies to the west.

29. EAST END OF CAVE.
You're standing in the eastern end of the ice cave. The cave is a dead end here, but you can make out a dark, shadowy area under the ice at your feet.

30. ROCKY TUNNEL.
You're standing in a small rocky tunnel which leads to the east. Above you there is a hole in the roof leading up to the ice cave.

31. ROCKY PLATEAU.
You are standing on a small rocky plateau with sheer drops on all sides. A tunnel leads into the rockface to the west. There is a Tardis standing to the east by the edge of the plateau.

32. TARDIS CONTROL ROOM.
You are in the control room of the Tardis. A small glass cylinder is sticking

up out of the console. It's the fluid link, and somehow the mercury has emptied out of it.

33. TARDIS CONTROL ROOM.
You are in the control room of the Tardis. The viewscreen shows that you are no longer on the rocky plateau: you seem to have shifted to a desert planet.

34. DESERT.
You are standing outside the Tardis on a desert world. The temperature is pretty unhealthy. However, you discover that the Tardis has landed underneath a huge spaceship which someone has carelessly parked here. An access ladder leads up into the craft.

35. DESERT.
As you wander off into the desert sands, your life support system throws in the towel, and you cook in your own juices.

36. CONTROL ROOM.
You are inside the control cabin of the spaceship, which seems to be a standard planet-hopper. You should have no trouble flying it, although the deep space radar has failed.

37-45. NEBULA.
The ship is flying through a dense dust nebula. You cannot see anything through the viewport, and the deep space radar is dead.

37. NEBULA (WHITE DWARF).
You are flying through a thick dust nebula. Although you can see nothing, the deep space radar shows a white dwarf star in the vicinity.

38. NEBULA (DUST).
You are flying through a thick dust nebula. Although you can see nothing, the deep space radar shows that there is nothing outside but dust.

39. NEBULA (PLANET).
You are flying through a thick dust nebula. Although you can see nothing, the deep space radar shows a planet in the vicinity.

40. NEBULA (BLACK HOLE).
You are flying through a thick dust nebula. Suddenly there is a lurch and the ship accelerates rapidly. Just before you and the ship are torn apart, you notice a black hole registering on the deep space radar.

41. NEBULA (RED GIANT).
You are flying through a thick dust nebula. Although you can see nothing, the deep space radar shows a red giant star in the vicinity.

42. NEBULA (ASTEROIDS).
You are flying through a thick dust nebula. Although you can see nothing, the deep space radar shows an asteroid field in the vicinity.

43. NEBULA (BINARY STAR).
You are flying through a thick dust nebula. Although you can see nothing, the deep space radar shows a binary star in the vicinity.

44. NEBULA (NEUTRON STAR).
You are flying through a thick dust nebula. Although you can see nothing, the deep space radar shows a neutron star in the vicinity.

45 NEBULA (SMALL MOON).
You are flying through a thick dust nebula. Although you can see nothing, the deep space radar shows a small moon in the vicinity.

46. REFUELLING DEPOT.
As you pilot the ship in for a landing, you pick up a navigation beacon. The ship lands at the solitary human base on the planet, a deserted, automatic refuelling depot. Your ship's tanks are replenished.

47. DISUSED LANDING PAD.
The navigation beacon on this world leads the ship down to an abandoned outpost which now consists mostly of wreckage. There is a clear spot in the centre where you bring your ship in to land.

48. WRECKAGE.
You are standing at the foot of the access ladder to your ship, surrounded by junk and wreckage. Most of it seems useless, but there is an aerial from a distress beacon which seems to be in fairly good shape.

49. CLEAR SPACE.
You've finally emerged from the dust nebula into clear space. You are circling a planet which appeared ahead of you as you exited.
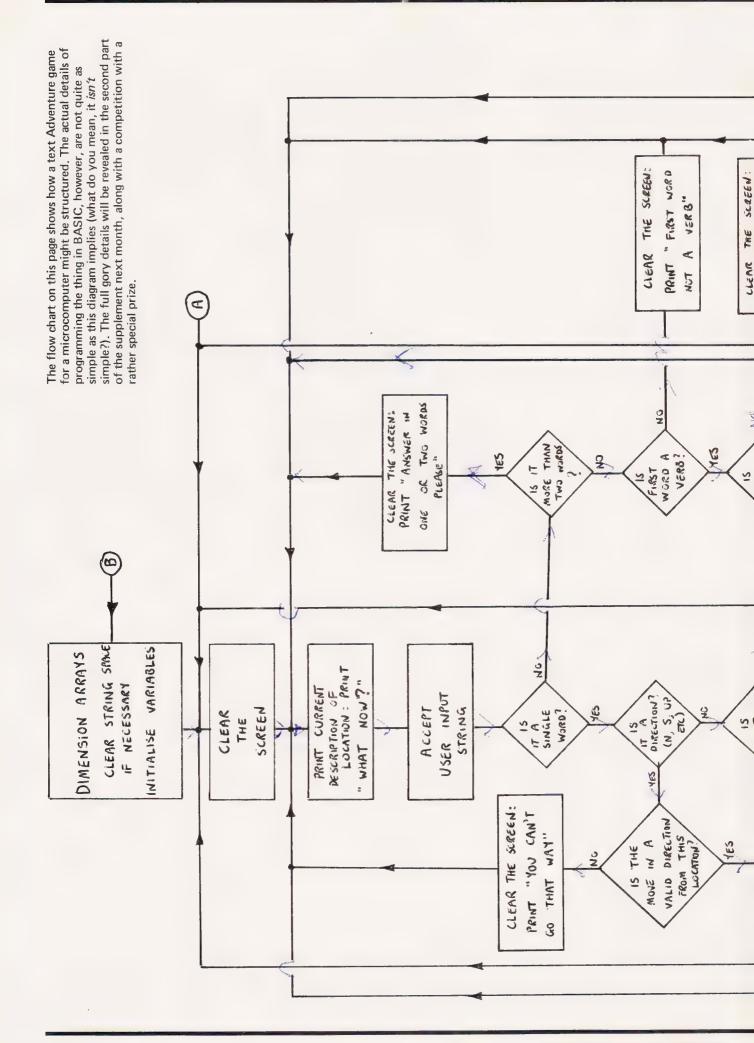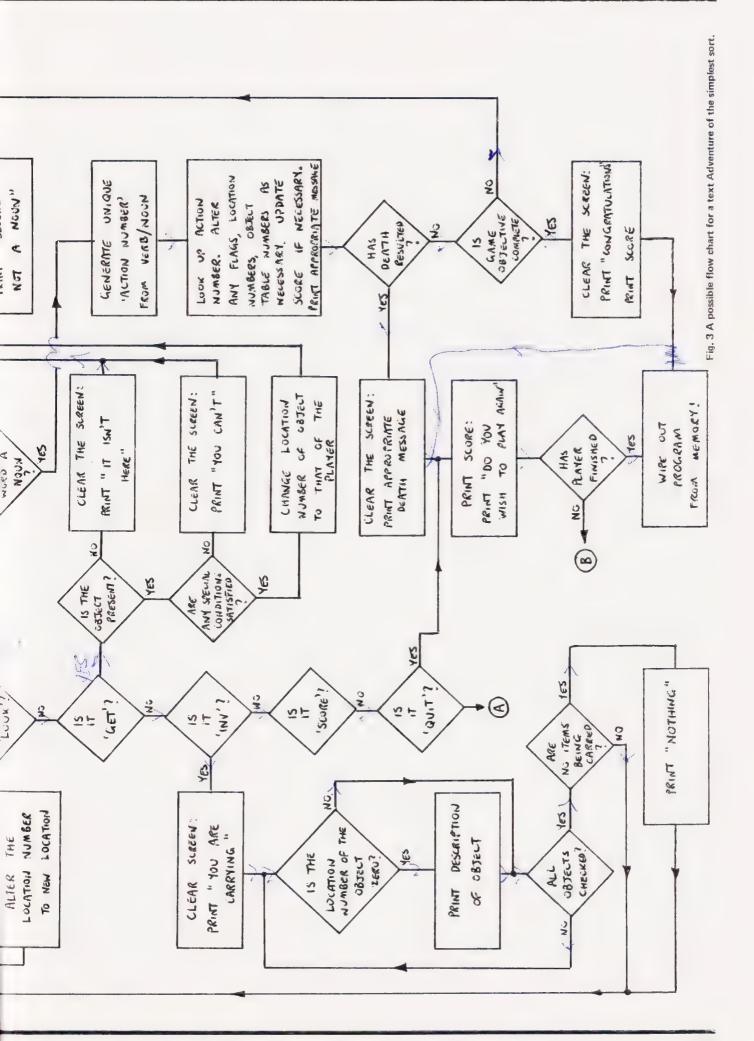
50. OUTPOST.
The navigation beacon on this planet leads your ship in to another abandoned outpost. This seems to be the slightly unfashionable end of the galaxy. However, things seem to be in a fairly good state of repair. The ship is sitting on the outpost landing pad.

51. DISTRESS BEACON.
As you reach the foot of the ladder you notice that the distress beacon situated on the landing pad is missing an aerial. Other than that it's in perfect working order.

The flow chart on this page shows how a text Adventure game for a microcomputer might be structured. The actual details of programming the thing in BASIC, however, are not quite as simple as this diagram implies (what do you mean, it *isn't simple?*). The full gory details will be revealed in the second part of the supplement next month, along with a competition with a rather special prize.

(B)

(A)

DIMENSION ARRAYS

CLEAR STRING SPACE IF NECESSARY

INITIALISE VARIABLES

CLEAR THE SCREEN

PRINT CURRENT DESCRIPTION OF LOCATION : PRINT "WHAT NOW?"

ACCEPT USER INPUT STRING

CLEAR THE SCREEN: PRINT "ANSWER IN ONE OR TWO WORDS PLEASE"

IS IT MORE THAN TWO WORDS ? — YES

NO

IS FIRST WORD A VERB ? — NO

YES

CLEAR THE SCREEN: PRINT "FIRST WORD NOT A VERB"

CLEAR THE SCREEN:

IS IT A SINGLE WORD? — YES

NO

IS IT A DIRECTION? (N, S, UP ETC) — YES

NO

IS THE MOVE IN A VALID DIRECTION FROM THIS LOCATION? — NO

YES

CLEAR THE SCREEN: PRINT "YOU CAN'T GO THAT WAY"

IS

NOT A NOUN"

GENERATE UNIQUE 'ACTION NUMBER' FROM VERB/NOUN

LOOK UP ACTION NUMBER. ALTER ANY FLAGS, LOCATION NUMBERS, OBJECT TABLE NUMBERS AS NECESSARY. UPDATE SCORE IF NECESSARY. PRINT APPROPRIATE MESSAGE

HAS DEATH RESULTED?

IS GAME OBJECTIVE COMPLETE?

CLEAR THE SCREEN: PRINT "CONGRATULATIONS" PRINT SCORE

WORD A NOUN?

CLEAR THE SCREEN: PRINT "IT ISN'T HERE"

CLEAR THE SCREEN: PRINT "YOU CAN'T"

CHANGE LOCATION NUMBER OF OBJECT TO THAT OF THE PLAYER

CLEAR THE SCREEN: PRINT APPROPRIATE DEATH MESSAGE

PRINT SCORE: PRINT "DO YOU WISH TO PLAY AGAIN"

HAS PLAYER FINISHED?

WIPE OUT PROGRAM FROM MEMORY!

IS THE OBJECT PRESENT?

ARE ANY SPECIAL CONDITIONS SATISFIED?

ALTER THE LOCATION NUMBER TO NEW LOCATION

"LOOK"?

IS IT 'GET'?

IS IT 'INV'?

IS IT 'SCORE'!

IS IT 'QUIT'?

(A)

(B)

CLEAR SCREEN: PRINT "YOU ARE CARRYING"

IS THE LOCATION NUMBER OF THE OBJECT ZERO?

PRINT DESCRIPTION OF OBJECT

ALL OBJECTS CHECKED?

ARE NO ITEMS BEING CARRIED?

PRINT "NOTHING"

Fig. 3 A possible flow chart for a text Adventure of the simplest sort.

WHATEVER HAPPENS IN THE FUTURE, IT'LL FIT INTO THIS SPACE.

MATTEL ELECTRONICS®
AQUARIUS™

PRINTOUT

Dear Sir,

Referring to a letter in your June '83 issue about saving/loading programs on cassette from a Casio FX 702P: I have an Akai GXC 310D which stores programs giving no trouble. Input specifications are 0.3 mV/5k: output specifications are 0.775 V (@ 9 dB VU) requiring 10k.

I record programs at −3 dB and play back at ¾ full volume. This method has worked on other decks as well (portable). Also, does anyone know the availability of RAM or ROM packs for Casio FX-702P calculators?

Yours faithfully,
B. Mutton,
Hamilton,
New Zealand

(*The best dealer we know of for Casio products is Tempus, 38 Burleigh Street, Cambridge CB1 1DG: you could write to them for a catalogue. That's in the UK, of course: we don't know too much about dealers in New Zealand! *)

Dear Sir,

I read the 'Next Month' page in your September issue with astonishment. Why? Well you mention the Sharp range of computers! Wow! This is a rare subject — very, very, very few magazines even know that a machine called Sharp MZ-80A even exists. But before, Mr. Editor, you put the brackets and asterisks around your reply, may I say that yes, Computing Today does know! Phew!

I have owned my Sharp MZ-80A since January and I am very impressed and discover new things regularly such as White Noise. I am only 17, but I am working on many projects concerning the Sharp MZ-80A. Things such as cartridge software, user-defined graphics, voice synthesis and writing my own fast interpreter/compiler. I also would like to start a Sharp MZ range user group; would anyone interested please write to me?

Could someone please tell my troubled brain how to access each dot in a character cell individually, find the code returned when Control is used with a key (pause for breath) and can the Character Generator ROM be easily replaced with a RAM? I have no connection with Sharp whatsoever.

Yours faithfully,
Roman Pestra,
51 Barrington Road,
London N8 8QT

(* Presumably you are after getting user-defined graphics on your machine, which is something Sharp seem determined not to provide (see the MZ-700 review last month). A hardware modification would be required, and we certainly wouldn't recommend how to go about it even if we knew the details! *)

Dear Sir,

I enclose a few lines of verse written straight from the heart and hot off the press! I am a severely handicapped spastic, chairbound, and with great problems, so I cannot write my name and must apologise in advance for my typed signature.

I assure you all the facts are there and the light-hearted nature of the poem (if one can label it poetry) only hides a distressing event behind a cheerful smile.

A TALE OF WOE

Hear the story of Jenny's plight,
Who bought a computer that
    wasn't right,
She thought she was covered by
    British Law —
Consumer Affairs and all that, you
    know —
So she took it back and demanded
    her cash,
To discover you shouldn't believe
    such trash.
They broke the rules, they'd sold
    her "short",
But Law is different and cheating
    "fair sport",
The Manual said it could not be
    had,
If lost, or broken, that was just too
    bad,
You were distinctly warned to take
    great care
Of the Master Tape — but it wasn't
    there —
There was only one with each
    machine,
So Jenny returned it — the
    computer, I mean,
And she purchased another in all
    good faith,
If that was correct, she was
    perfectly safe.
However, the salesman, in spite of
    the rule,
Produced a second — she'd been
    a fool.
How he achieved it, nobody
    knows,
But that is life — that's how it goes
The Law had been met, and the
    Law is tough.
Honesty simply wasn't enough —
The guilt was there, if not the
    intent,
But until it was proved they were
    innocent.

The Sharp MZ-80A: not a figment of the imagination.

She couldn't plead for common
   sense,
Nor could she afford any more
   expense.
This is her story, this is where she
   stands —
She has **two** computers, but **one
   pair of hands!**

Yours faithfully,
Jenny Waller,
Birmingham

Dear Sir,

We would like to use your Printout
column to inform Commodore
users of a chance to get hold of
some of that elusive American and
Canadian software, especially
those with the Commodore 64.
   We are now in contact with the
Toronto Pet User Group (TPUG).
TPUG have a library of over 3000
programs available for all
Commodore micros from the VIC
20 to the Super Pets. Users in this
country can obtain access to the
TPUG library by becoming
overseas members; this gives them
a newsletter 10 times a year plus
new software every month, and of
course full access to the TPUG
library.
   What we need is at least 60
Commodore users to join us as a
group in order to get block
membership at reduced rates. We
will be happy to organize the
whole thing, and produce a
separate supplementary newsletter
on a regular basis.
   Anyone interested should
contact us, giving details of their
micro and we will get in touch
with full details of the whole thing.
   I would like to emphasize that
unless we get 60 users then the
price of membership would be
much greater, so you Commodore
users let's be hearing from you.
   I would like to thank you in
advance for publishing this
letter, and thanks for one of the
best mags around (creep).
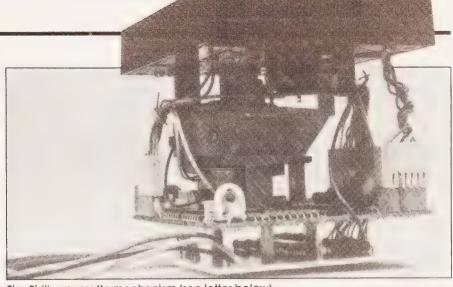
Yours faithfully,
Peter Paton,
Patons Computer Services,
176 Todmorden Road,
Burnley,
Lancashire BB11 3EU

Dear Sir,

With reference to your special
report on Page 55 of the July issue
of your magazine, you state that
the Hobbit drive is the first time
that the Philips digital cassette
mechanism has been implemented
in a complete operating system.
We beg to inform you that this is
not the case, as dual Philips digital

The Philips cassette mechanism (see letter below).

cassette mechanisms have been
integrated into our Brailink
Computer Terminal for blind
operators since 1980.
   You may be interested to know
that after considerable experience
of operating this system, we are
also looking at the personal
computer market with a range of
new operating systems.

Yours faithfully,
J. Glazer,
Clarke & Smith Manufacturing
Co Ltd,
Surrey.

Dear Sir,

I own a Cortex and would rather
have bought a BBC on the whole.
   Can you tell me what is the
point of a 12 MHz clock when the
chip simply divides it by 4 to
effectively run at 3 MHz? Is it used
for some sort of microcode
sequencing or just to look good on
paper? The TI 9900 Family Data
Book gave no indication
whatsoever.
   Also, can you tell me how well
is Cortex doing in terms of sales?
By the delay in getting accessories
I get the impression that the
Cortex is being snatched up like
hotcakes, but I don't notice anyone
confessing to having one
anywhere.

Yours faithfully,
C.C. Kuan,
Hendon.

(★The TMS9995 microprocessor
requires a four-phase internal
clock to operate correctly, and the
easiest way to get that accurately is
to divide down a faster clock by
four. Since you have the 9900 data
book, look at the timing diagrams
on page 1-14 to see what's going
on.
   Sales of the Cortex have been
pretty healthy, we understand, but
there have been delays in delivery

caused by component supply
problems. If anyone out there has
any software written for the
machine, we'd be glad to consider
it for publication in CT ★)

Dear Sir,

What, I wonder, as I sit before my
NewBrain, is machine code all
about? What do I type into my
QWERTY keyboard to make that
jump through BASIC to meet the
chip on its own level? (I could
write to Grundy, but I think I'll
save the stamp, and 50 quid for
their software handbook is just not
on).
   There must be others like me,
not computing experts, but
wanting more than trivial arcade
type games. Let's be honest, no
one needs a home computer; you
can work out your mortgage on an
envelope and an address book is
data base enough. Surely
computing is for the joy of
computing itself, and machine
code holds such promise, if only I
knew how! Could you therefore
recommend an Idiot's guide to
Machine Code, or better still put
something about it in CT (I am a
subscriber, after all), but not too
much theory please, I want to
make use of it, not just have a
vague idea about pointers and
flags and things.

Yours faithfully,
Giles Becket,
Vlaardingen,
The Netherlands

(★ The two best books for Z80
programming that we know of are
Programming the Z80 by Rodnay
Zaks, published by Sybex at
£11.50, and **The Z80
Microcomputer Handbook** by
William Barden, published by
Sams and available for £10.95
from, amongst other outlets, the
Book Service of our sister
magazine Electronics Today
International. ★)

CHOOSING A HOME MICRO

Choosing a home micro can be a daunting task to the newcomer, and with an ever increasing number of micros emerging on the market, even up-grading, say, from a ZX81 can be a risky and expensive exercise if the wrong decision is made. It is important to look at the real facts and specifications, and check exactly what you get for your money before choosing your micro-computer system.

THE PITFALLS

"DON'T LET THE ADD ONS ADD UP"

A number of large companies are offering packages that seem to be good value and low cost.
These offers usually have a hidden sting inasmuch as the essential accessories such as connection leads, peripherals and software often carry very high cost premiums.
e.g. software for low cost hardware usually costs between £29 and £49 for a ROM cartridge!!

CHECK THE QUALITY OF THE PRODUCT.

Raw materials are now an area where corners can be cut, and shoddy workmanship during 'building' can effect the 'up-time' of your unit. Areas to watch out for are unreliable edge connectors, corrosion and poor quality P.C.B.s. Low quality components and bad design will seriously effect the reliability of the end product, and can lead to false economy.

DON'T BUY A GAMES MACHINE

Unless you want just games and nothing else! With a games computer you are limited.
Some computers, however, have the advantage of both games facility plus the whole world of computing to explore, as your interest and skills develop. A real computer system will allow you to expand your knowledge of the Hi-Technology world, and help earn its keep with its added uses in the field of education, communication and home business use.

SOFTWARE

Make sure the system you choose has a growing library of support software, to enable you to realize the full potential of your machine.

KEY POINTS TO LOOK FOR

● High Resolution Colour

In general most home computers have a poor graphics resolution (or detail). Check on the vertical and horizontal resolution in graphic mode and multiply the two numbers together. If the result is less than 35,000, then the graphics can hardly be considered high resolution. Without high resolution graphics displays such as those used in games tend to be "Chunky" in appearance.

● High Quality Sound

Some computers claim to provide a sound channel when in reality all that can be found inside the computer is a small buzzer controlled by electronic pulses. At the very least a sound facility should provide more than one channel and a raise channel as well (for gun shot effects in games for example). The best systems also provide envelope control of the sound channels to produce very sophisticated effects; very important for generating music. Also look for the ability to connect to external amplifiers.

● Keyboard

For accurate entry of programs and data into a computer it is important that the keyboard has a good tactile feel in operation. Coupled with acoustic feedback the user is fully aware when the computer has accepted his/her actions. Also of importance in a keyboard is layout. A standard computer keyboard layout will familiarise the user with the vast majority of computers used in the world of business and professional applications; very important if the purpose of purchasing a computer is educational.

● RAM

One of the most important features of a computer is the amount of RAM, or memory, included. In general the more powerful and exciting a computer program is the more RAM it requires. But take care, all computers are advertised quoting the total RAM used in the system. Computers use up a great deal of their own RAM for storing essential data and particularly in supporting the graphics display and the CPU. If it is less than 32K think again. is it enough?

● Computer Language

It is too dificult to program a computer in its own binary language so high level languages are used, the most popular being BASIC. However, there are a number of BASICs. some being very different from the rest. A de facto standard in the computer industry is Microsoft BASIC. Learn this one and you will be able to program in the majority of computer BASICs; such an important point if a home computer is to be used to educate your children to face the technology of the future.

● Expansion

As your interest and knowledge of computing grows, you will need a

Choosing the right system carefully will save you from throwing your money away. Check full specification, plus peripherals and software prices, before you buy. Preferably choose a Real computer system that can expand to meet your needs.

computer system that will grow with you; able to accommodate Printers. Disk-drives. Joysticks, Communications Modem. and Colour Monitor, as well as produce HI-FI sound effects.

● Software

The computer you choose should have a growing selection of utility software to make the most of its capability.
Remember, computing is here to stay. You can't learn to compute on a toy, or a device which does not behave like a real computer.
In short, look out for a computer which offers all the points above, and you will be sure of getting the best value for money

To find out which company offers you the right choice, with:-

● Good value, high specification, quality micros.

● A quality, 4 colour, plain paper printer/plotter.

● Communications Modem.

● Micro Disk Drives.

● Comprehensive and growing range of software

TURNOVER... ➡

Paul Gardner

LEARNING FORTH

With the advent of so many implementations of FORTH for small microcomputers, it's about time we ran a teaching series for those who want to learn a second language. So here's part 1.

Hardly a month passes by without another version of FORTH being released for yet another new micro. This once led me to wonder what FORTH is and why it appears to be so popular. This series of articles will attempt to show what makes up a FORTH system, how and why it differs from other languages, such as BASIC and Pascal, and what its advantages and disadvantages are when compared to such languages.

WHAT IS FORTH?

For a start, FORTH is not new; it was developed in the late sixties by Charles H. Moore. At the time he was working on an IBM 1130, a 'third generation' computer. He believed the results of his efforts to be so powerful that he considered it a 'fourth generation computer language', and would have called it FOURTH except that the IBM 1130 permitted only five-character identifiers. Thus FOURTH became FORTH, so now you know!

FORTH, according to its creator, is many things at the same time. It is a high level language, an operating system, an assembly language, a set of development tools and 'a software design philosophy'. If you consider the operation of a traditional mainframe computer, it consists of several elements, all under the control of an 'operating system' which allows users to converse with the machine, to input text, write, compile and run programs, use previously written application packages and so on.

For each portion of the whole system a significant amount of learning is involved. First you have to learn how to converse with the computer? this involves knowing a little of the operating system. Then if you want to write programs you need to know at least one language, then how to user the editor to correct your mistakes, how to compile and run your programs, and so on. Very often each element of the system will have what you might call a different 'protocol' so that it may be a long time before you can have a program up and running.

A FORTH operating system will do all that a traditional system will: however, only the bare minimum will be written in the hardware-dependent code of the machine, and the rest will be written in FORTH. Thus the entire system observes a similar 'protocol' and is much easier to understand and learn. Because the FORTH operating system is much simpler than its traditional counterparts, it can run much more quickly, more conveniently and in much less memory.

"This is all very well", you may say, "but what has FORTH to offer to micro users?" Well, FORTH is *fast*. A quick comparison between Sinclair BASIC and Abersoft's FORTH shows that FORTH runs about 30 times quicker than BASIC on the Spectrum. While microcomputer versions of FORTH are not remarkably fast compared to machine code (about 10% of machine code speed), it is very simple to write time critical sections of a program in assembler to run at full processor speed. Even if your version of FORTH does not include an assembler, machine code sections can usually be implemented much more easily than on a BASIC system.

FORTH is very compact — the entire operating system and its word set occupies less than 8K.

FORTH is transportable; about 90% of a FORTH system is written in FORTH itself and this makes it relatively easy to implement on almost any machine. This leads to an important point — FORTH sticks to a well defined standard (well, two actually). Most versions of FORTH are written to FORTH-79 or fig-FORTH standards and the differences between these two are minimal. This means that a program written for one machine can easily be adapted for another. Compare this with most BASIC programs.

Finally, FORTH is infinitely extendable. To write a program or application in FORTH, you simply define new commands in terms of the commands present in the initial FORTH system. This idea covers every aspect of the language, so that if you want to produce a new maths routine or even extend the compiler to produce a few more structures, the method is the same.

A FORTH SYSTEM

This series of articles is based around Abersoft's version of fig-FORTH for the Sinclair ZX Spectrum, although the ideas apply to all versions of FORTH and most of the listings given can be implemented on any machine. Abersoft's FORTH (also reviewed in Three FORTH's on page 76) arrives as a cassette tape and a manual which gives details of the implementation, the text editor and the extensions provided for the Spectrum, such as screen manipulation commands.

The system can be considered as comprising of four parts:

● The Dictionary
● The Keyboard Interpreter

- The Stacks
- The Disc

The dictionary, as the name implies, is a list of words which can be regarded as commands and associated with each word is a set of instructions which are executed whenever the word is encountered.

Assuming you have loaded your version of FORTH, type:

VLIST   (and then press Enter)

The characters you type appear on the screen and are also sent to a keyboard buffer. When you press Enter the contents of the buffer are executed by a routine called the 'keyboard interpreter'. In this case the keyboard interpreter takes the word VLIST and attempts to look it up in the dictionary. When it finds the word, the instructions associated with it are executed, in this case, 'display the words contained within the dictionary'; and so it does.

The manual gives an explanation of each one of these words, but you need not attempt to understand many of them for now.

You can type in more than one word at a time, but each word must be separated by at least one space, for example:

VLIST VLIST VLIST (Enter)

If you get a little bored, press CAPS SHIFT and 1 to Break.

If you look at the end of the list of words you will see that the computer has responded with 'OK' after executing your instruction. It does this whenever it has completed executing all the instructions in the keyboard buffer. If you simply type nothing and press Enter the computer responds 'ok' to let you know it has executed all the commands given (ie none) and is awaiting more.

SIMPLE DEFINITIONS

As I have said, you can extend the dictionary to include words of your own devising. Here is a trivial example:

: GREETING CLS CR ." HELLO THERE" CR ;

You will need to type this in all in one go before pressing Enter unless you know how the editor works.

If you mistype and press Enter the computer will respond with an error message; for the moment ignore this and try again (but use a different name, like GREETING1).

Assuming all goes well and the computer responds 'OK' then typing:

GREETING (Enter)

will clear the screen, print "HELLO THERE" on the second line and respond 'OK' on the third. If you now execute VLIST you will see the new word at the beginning of the dictionary list.

This trivial program demonstrates quite a lot of the FORTH system. As I have said, when you type into the keyboard buffer and press Enter the interpreter executes your commands one word at a time. In this case, the first word was : which means 'set up a new entry in the dictionary using the next word in the buffer as the name, and compile the following words as the instructions associated with the word'. The word ; ends the definition.

You will notice, if you try, that you cannot list GREETING as if it were a BASIC program; instead you get an incomprehensible message which I will explain later. This is because your definition of the word GREETING has been compiled by the FORTH compiler and added to the dictionary. Later I'll show you how to save a copy of your definition.

If you're fed up of being greeted, try another word:

: BRUSHOFF GREETING ."
CAN'T STOP, MUST RUSH . .
BYE" CR ;

You can see from this example that once you have defined a word such as GREETING you can use it in the definition of another word, as in the definition of BRUSHOFF above. The idea of writing many small definitions and then using them in later definitions is fundamental to FORTH programming and does take a little getting used to.

THE STACK

Try this: type in three numbers separated by spaces and then press Enter, for example

10 20 30

The computer responds with 'ok', but what has happened to the numbers? I have said that whenever you press Enter the keyboard interpreter attempts to look up and then execute the words in the dictionary and that if they are not in the dictionary, responds with an error message. Well, the whole story is that if the interpreter cannot find your word (for example, 10) in the dictionary it attempts to treat it as a number. If it is a valid number then it is put into an area of memory called the 'stack', and there it is left. The stack is a method by which

the computer can store numbers to be used later. It behaves in the same way as coins put into a long tube — the ones put in first go to the bottom while the ones put in last stay at the top.

To recover your three numbers type

. . . (Enter)

The word . (pronounced dot) means 'take the top number off the stack and put it on the screen'. You will see that the number entered last comes out first and vice versa.

If you type . again you will get an error message meaning 'stack empty'. Although a number will be printed, this is a number contained in the region of memory below the stack and is of little significance.

This brings us to the way FORTH deals with numbers and performs arithmetic. Try typing:

3 4 + . (Enter)

The computer responds

7 ok

which is undoubtedly the correct answer. You will see that the two numbers you typed come before the plus operation. The type of notation FORTH uses is called 'post fix', ie the operator comes after the operands. So the command + (plus) means 'take the top two numbers off the stack, add them together and return the answer to the top of the stack'.

All FORTH arithmetic commands work like this: the operands they are to perform upon must be on the stack before the operator is used. Here are some examples, with your entry on the left and the computer's response on the right:

3 4 * .	12 ok
20 8 / .	2 ok
9 12 - .	-3 ok

You will see from the last example that negative numbers behave quite normally, but that the division operation gives an answer of 2 and not 2.5. This is because standard FORTH works in integers only (and for very good reasons), so that 20 8 / . gives an answer of 2 and discards the remainder.

There are other commands which will take values from the stack and return not one but two answers. Try:

20 8 /MOD (Enter)

The response will be:

2 4 ok

ie 20 divided by 8 gives 2
remainder 4.
There are many more
operations which I will go into later,
but for now here's how to define one
of your own. There is no command
in standard FORTH to square a
number fo you need to write one
yourself:

```
: SQUARE DUP * ;
```

Once written this behaves as any
other FORTH command. In this
case it requires one number on the
stack, and will return the square of
that number. For example:

```
2 SQUARE .        4 ok
8 SQUARE .        64 ok
```

An interesting word, DUP, appears
in the definition of SQUARE. This
means 'make a copy of the top
number on the stack and put the
copy on the stack'. So:

```
2 4 DUP . . . (Enter)
```

gives 4 4 2.
So DUP (pronounced dupe) has
duplicated the top number. (If only
I could do that with my tube of
coins.) Notice that the last number
onto the stack was 4, so that's what
we get two of.
There are many more 'stack
manipulation' commands, but first it
is best to know how to write some
slightly larger definitions and store
copies of them.

THE DISC

Up till now I have given small
examples of how to define a new
word by simply typing in a word
like GREETING and compiling it
immediately. This is okay for short
trivial examples, when you can just
do it again if you make a mistake;
however, for even the simplest
applications you need to be able to
store your original defintion in its
'source' code so that you can more
easily change the word and compile
it again.
Traditional FORTH systems use
a disc drive to store written text
which can later be compiled. The
Abersoft version of FORTH,
however, sets aside 11K of the
Spectrum's memory to 'pretend' to
be a disc drive. The operating
system can communicate with the
'disc drive' through a region of
memory called the block-buffers,
instead of dealing directly with the
'disc'. Although 11K may sound like
a lot of memory to take up with a
pretend disc, this still leaves you
18K for your own FORTH

definitions which, believe me, is
plenty.
To put information onto the
disc, you have to use a set of
FORTH words which comprise the
EDITOR. To use the editor, first
type:

```
EDITOR
```

You can now list and change any
'page' of the disc, using a set of
commands which should be defined
in your manual. (The editor does
not need to keep to any standard,
but most are similar.) To start with,
type

```
1 CLEAR
```

This selects page one of the disc
and erases the previous contents (if
any). As with the arithmetic
operations, you need to have the
page number on the stack before
CLEAR is executed. Now type

```
1 LIST
```

This will display page one of the
disc. In the Abersoft version the
11K RAM disc is divided into 11
pages numbered 0 to 10, each of 1K
bytes which is divided up as 16 lines
of 64 characters. This,
unfortunately, is not how the
Spectrum screen works but, as you
will see, it does quite well.
At the moment you should see
an almost empty screen, with 16
lines numbered 0 to 15 down the
left-hand side. To input text to the
disc, try typing the following:

```
0 P THIS IS HOW TO PUT (Enter)
```

```
1 P TEXT ONTO THE DISC (Enter)
```

If you now type

```
1 LIST
```

you will see that your two lines have been added to that page. The word P stands for PUT and 1 P, for example, means put the following text onto line one of the current disc page.

There will be full instructions in your manual about how to put information onto the disc, and it is wise to get used to the editor before trying to write any long definitions.

The disc memory can store any sort of text, from a FORTH definition to a letter to your Mum. The disc is only to store the information, what it is for does not matter until you come to compile it (if it is a definition).

As a quick example of how to write a definition onto disc and then compile it, try the following example. First of all, type COLD to clear all the definitions you have written so far. Type EDITOR to enable you to use the disc, then 1 CLEAR will empty page one of the disc. Now enter Listing 1 exactly as shown. Note: there *must* be a space between the open bracket and the first letter of the title, ie (LARGE.

Figure 1 shows a listing of this page of the disc. (If you have a ZX printer attached, type 1 LINK and everything that appears on the screen will be sent to the printer. Type 0 LINK to stop the printer.) Figure 2 gives a brief explanation of the action of each new word.

Hopefully you have managed to edit the page to look like the listing. Now type

1 LOAD

and the computer will whine for a bit before responding 'OK'. What has happened now is that the contents of page one of the disc have been compiled in much the same way that you can compile small definitions from the keyboard, only now if it does not work you can list page one to see your original text.

If you get a message during compilation then there is a mistake in your disc text; if you VLIST then you can see the name of the definition in which the mistake was made. To clear an unwanted definition from the dictionary type FORGET (name) where (name) is the name of the definition you want to remove. BEWARE — FORGET will remove that definition *and all those that come after it in the dictionary*.

If during compilation you receive a message like

STAR MSG # 4

this means that there is already a word with that name in the dictionary. Using Abersoft's FORTH, this will not stop the compilation, so it is only a warning.

If you type F now, the screen will display a large letter F almost centrally. Not very exciting, I know, but we are only beginning.

Once you have written a successful version of an application onto disc you can save a copy of the whole RAM disc using the command SAVET. This can later be reloaded using LOADT.

THE STORY SO FAR

So far I have mentioned the dictionary; simple definitions entered form the keyboard; the stack; FORTH arithmetic — post fix style; the disc (or RAM disc, for the poor amongst us); the editor, and how to compile definitions from the disc.
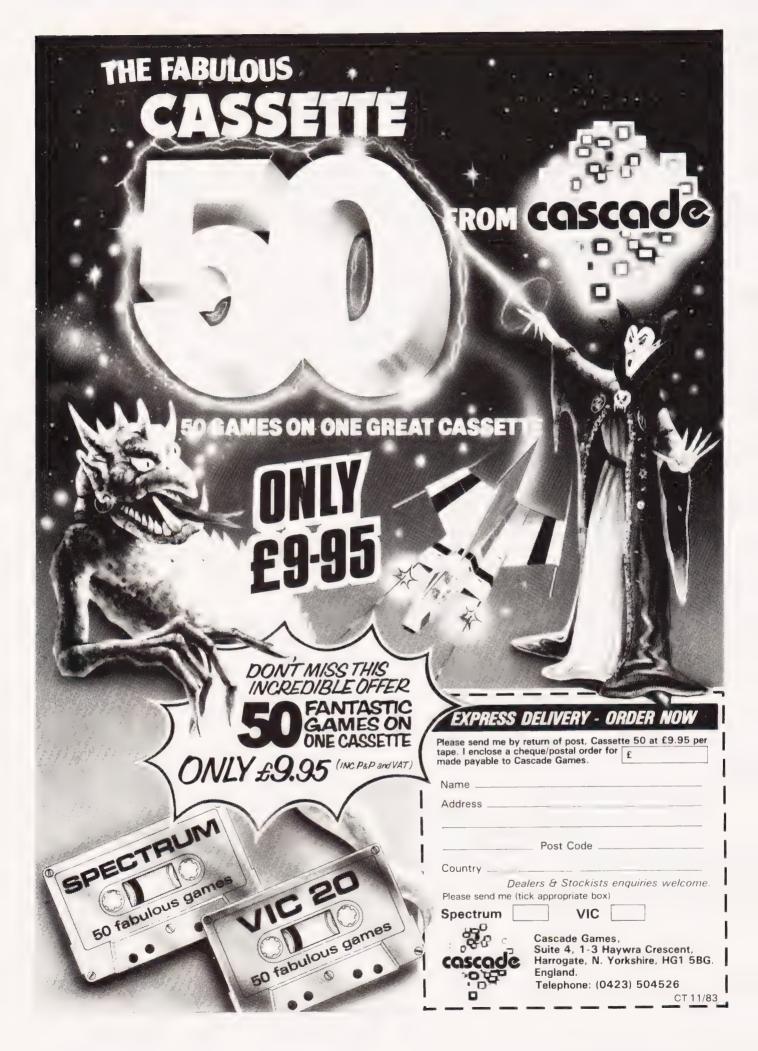
If you haven't already got a version of FORTH for your machine, then buy or borrow one for a while and have a play; once mastered it is a very addictive language. Until next month I suggest you get to know the editor, as it will make life much easier in the future.

A very useful book is **Starting FORTH** by Leo Brodie, published by Prentice Hall, ISBN 0-13-842922-7 (paperback). Otherwise a copy of the Jupiter Ace manual, **FORTH Programming**, by Steven Vickers is worth having a look through, and is available separately from many bookshops.

```
0  P  ( LARGE LETTER-F )
1  P  : STAR 42 EMIT ;
2  P  : STARS 0 DO STAR LOOP ;
3  P  : MARGIN CR 10 SPACES ;
4  P  : BLIP MARGIN STAR ;
5  P  : BAR MARGIN 5 STARS ;
6  P  : F BAR BLIP BAR BLIP BLIP CR ;
```

Listing 1. The 'Big F' program. Type it in carefully as spaces are important.

```
SCR # 1
  0  ( LARGE LETTER-F )
  1  : STAR 42 EMIT ;
  2  : STARS 0 DO STAR LOOP ;
  3  : MARGIN CR 10 SPACES ;
  4  : BLIP MARGIN STAR ;
  5  : BAR MARGIN 5 STARS ;
  6  : F BAR BLIP BAR BLIP BLIP C
R  ;
  7
  8
  9
 10
 11
 12
 13
 14
 15
ok
```

Fig. 1 A listing of page 1 of the disc.

```
SCR # 2
  0  LETTER-F GLOSSARY
  1
  2  NAME      DESCRIPTION
  3  STAR      PRINTS AN ASTERISK
  4  STARS     TAKES A NUMBER OFF
  5            THE STACK AND PRINTS
  6            THAT MANY ASTERISKS
  7  MARGIN    PRINTS NEWLINE AND
  8            10 SPACES
  9  BLIP      PRINTS MARGIN AND AND
 10            AN ASTERISK
 11  BAR       PRINTS MARGIN AND 5
 12            ASTERISKS
 13  F         PRINTS A LARGE LETTER
 14            F USING BAR AND BLIP
 15
ok
```

Fig. 2 Page 2 of the disc describes the program.

NO LIMIT

Now, a home computer with virtually no limit to its possibilities. The astonishing new Sharp MZ700. A machine with a dazzling array of talents.

First, it's a 'clean' machine. So you are not limited to any one computer language. You have the flexibility to run and write programs in BASIC, FORTRAN, MACHINE CODE, PASCAL, ASSEMBLER and many others. And the MZ700's 512 predefined characters mean you can build up detailed pictures on the screen, without spending time specifying and designing special characters for games and special effects.

Second, it has a memory of 64K. So as your technique improves and develops, you are able to move forward to more and more advanced programming.

All of which makes this the perfect home computer for parents, as well as children.

The MZ700 gives you access to a wide choice of new software, from only £3·95 per cassette. An additional plotter/printer, costing £129·95, can produce high resolution graphics in 4 colours. A data cassette recorder is an extra at £39·95. Both add-ons fit snugly into this easy to carry compact system with no trailing wires. And you get ten exciting games, free on purchase, including Super Puckman, Circus Star, Snake v. Snake, and Man-Hunt.

The brilliant new MZ700.
The no-limit computer. £249·95. From Sharp. Where great ideas come to life.

CPU:	Z80A	1
ROM:	Monitor 4K byte ROM	1
	Character generator 4K byte ROM	1
RAM:	64K byte D-RAM	8
	4K byte V-RAM	2
I/O bus:	Expansion I/O bus	1
	Additional printer I/O bus	1
	Cassette READ/WRITE terminals	2
	Joystick terminals	2

RGB MONITOR/VIDEO MONITOR/RF OUTPUTS

COMPATIBLE
WITH SHARP MZ 700 SERIES

Look for this logo on
software denoting
Sharp Compatible Software.

The world of
SHARP
where great ideas come to life.

Don Thomasson

THREE FORTHS

Implementations of FORTH for various popular computers are now appearing thick and fast. We review three of them, two for the Spectrum and one for the Oric.

The advent of the first personal computer working entirely in FORTH, the Jupiter Ace, seems to have triggered off a spate of FORTH implementations for other small machines. Some, like the Ace itself, follow the FORTH-79 standard, but the three examined here are based on fig-FORTH. They are:

Abersoft Spectrum FORTH
Artic Spectrum FORTH
Oric FORTH

Since they all derive from the same source, the documentations provided by the FORTH Interest Group, it is natural that the three should have much in common, but there are quite important differences, which begin to become apparent when the programs are loaded.

LOADING

The Abersoft tape loads in about 70 seconds, and programming in FORTH can start immediately, with all facilities available.

The Artic tape loads in about the same time, but whereas the Abersoft tape can be left to load itself while coffee is brewed, the Artic tape must be watched and stopped when the load is complete, as there are three more files to be loaded, these containing the source code for the EDITOR.

To set up this essential extension, you type in 1 LOAD and run the tape when instructed, stopping smartly when the load is complete, as the data that has been read in must be compiled. This is repeated for the other two files, and the system is then ready for use.

The Oric tape, recorded at slow speed to minimise problems of recorder compatibility, takes an age to load, but this can be avoided by re-recording the program at fast speed. A block of seven files must then be read in in a single operation, these setting up the EDITOR source code. When this has been compiled, there are 16 other files which can be loaded and compiled to add optional extras to the FORTH vocabulary.

Both the Spectrum systems would benefit from adaptation to the Microdrive, the Artic version becoming far more practicable, and all three would benefit from the availability of cassette motor control, but as they stand the Abersoft implementation has a clear lead at this stage.

PROGRAM STORAGE

The difference in loading procedures are linked with the method used to store program source data. FORTH was originally developed for use with disc systems, which can supply blocks of fresh data rapidly. Source code was held in data blocks called 'screens', each screen holding 1 kilobyte of data. The screens were called up from disc as required, with perhaps a 'cache' of recently-used screens held in RAM for immediate reference. If a fresh screen was called up when the cache was full, the 'oldest' screen was put back on disc, if it had been altered, and the new screen took its place.

Replacement of discs by a cassette recorder necessitates a change of method, and a measure of compromise.

The Artic system remains close to the original concept, but with no cache. Only one screen can be held in memory at a given time. When a new screen is needed, it has to be brought in from tape, the previous screen being saved first if it has been updated. Each screen can be identified by a number from 0 to 32768, but searching for one screen among even 10 recorded files can be tedious. A recorder fitted with a counter is essential to allow a given screen to be located quickly, and to ensure that when it is re-recorded it will not erase any other files. In the absence of motor control, the process demands constant attention.

A modified version adapted to work with Microdrives would be much more friendly, but the adaptation would not be simple.

The Oric approach provides a cache of seven screens. A specified number of consecutive screens can be saved or loaded in one operation, but the tape counter is again almost essential to the process of locating existing files or free tape. The cache steals 7K of RAM, roughly speaking, but leaves ample room for dictionary extensions.

The Abersoft system provides a cache of 11 screens, which are saved and loaded as a single file. This leaves more than 18K bytes of space for dictionary extension, which would take quite a lot of filling, and the only snag is that all the tape files have the same name — DISC. However, as they are fairly large files, all of the same length, finding the one you want is not too difficult. It is possible to transfer or copy a given screen from one file to another by saving it in a ghost screen, ie one with a negative number, but that is not an advertised feature.

A version for use with the Microdrive is promised, and this should remove the last impediment to simple use of the system.

VOCABULARY

The words defined in a fig-FORTH dictionary are not rigidly specified, on the basis that it is not too difficult to add extra words if the need for them arises. However, addition of some words involves machine code programming, and some users would find this a serious problem.

Artic provide CIRCLE and COPY, which are not in the Abersoft version. On the other hand, Abersoft provide about 40 words which are missing from the Artic dictionary, and these include several words needed to handle double numbers and to output numbers as unsigned. Not one of a number of programs which had been written for the Abersoft version could be run on the Artic FORTH without supplying the missing definitions.

The Oric version also lacked provision for handling double numbers properly, but supplied some extras, such as PICK, which delves into the depth of the stack to find numbers and other data buried there. The Oric system implements the valuable CASE structure as an optional extra, the Abersoft system includes it as standard. The Artic system omits it altogether.

The sad thing is that some users may never discover that the missing words exist, and some of the capabilities of FORTH will remain a closed book to them.
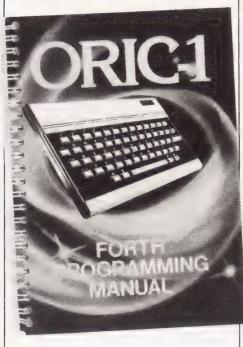
The biggest difference in the facilities offered has been left until last: the Oric version provides an assembler as an optional extra. This is useful for anyone with knowledge of machine code, and it seems a pity that no Z80 assembler is yet available in FORTH, though Abersoft aim to produce one, and also to implement floating point. As matters stand, all three work in 16 and 32 bit integers, though the facilities for 32-bit working are limited in the Artic and Oric versions.

It should be remembered that extensions of this kind can be supplied as source code to be added to the existing dictionaries, the modified system then being saved on a new tape. On the other hand, it is not usually possible to modify existing dictionary entries, which have to remain dormant when replacements are added.

All the EDITOR systems in the three versions are nearly identical, though the Artic system provides a copy facility based on use of the arrow keys.

MANUALS

Documentation of FORTH is always a problem, as some of the words are likely to become confused with punctuation marks, and this has proved a difficulty for all three implementations. Errata sheets would be desirable, because the meaning of FORTH words needs to be understood clearly. It is also disconcerting when unlisted error numbers appear, perhaps because HEX working is selected. It would be so simple to insert the word DECIMAL into the definition of MESSAGE, or at least warn the user...

The Oric manual was the largest, and the most confusing, Chapter Six having apparently slipped down the gap between pages 24 and 25. As it was billed to cover program and data storage, its absence was not seen as trivial.

The Artic manual gave most information, but even that constituted no more than a bare introduction to the language. Learning to make the best use of FORTH takes time and patience, and too many books on the subject take you over the first few hurdles, then leave you on your own. (That's why we're running a new teaching series, starting this issue. Ed).

USING FORTH

At this point, it is opportune to take a broader view. Is FORTH really worth bothering about? One view was that the study of almost any other language would be more profitable, but that was the view of 'professional' professional, who had a typically biased attitude.

The value of FORTH depends on the way it is used. It offers economy of time and space, and if those factors are unimportant another langauge may be preferable. Potential machine code writers are often deterred because of the difficulty of writing input/output routines, mathematical processes and the like. By linking their code to FORTH, they can solve the problem.

On the whole, it can be said that everyone interested in computers should at least investigage FORTH. This would be done most effectively on a machine able to display a complete screen at one time, the smaller screens of the Spectrum and Oric tending to overflow when the FORTH screens are full, but that is a minor point at the investigation stage. These programs, flawed or not, will open up an interesting new world.

The version which emerged from the review with most credit was clearly that provided by Abersoft. By comparison, the Artic version was distinctly unfriendly and limited in scope. The Oric version would benefit from inclusion of some extras in the basic dictionary, and the addition of the words needed for double and unsigned numbers.

It would, perhaps, be fair to suggest that few current FORTH implementations are fully mature, and as the needs of users are better understood there may be more extensions to the standard dictionary. Meanwhile, it is possible to begin to study the language.

48K Spectrum FORTH
Abersoft,
7 Maes Afallen,
Bow Street,
Dyfed,
SY24 5BA

FORTH
Sinclair Research,
Stanhope Road,
Camberley,
Surrey,
GU15 3PS

ORIC-FORTH
Tansoft Ltd,
3 Club Mews,
Ely,
Cambs,
CB7 4UN

Don Thomasson

PROBLEM PAGE

In Part 4 we work backwards to a solution of last month's puzzle, and set a problem about sets.

Last month's puzzle involved computation of N, where N is given by the equation below (a mathematical representation of the puzzle's structure):

$$(((((((N/2) + 6)/2 + 5)/2 + 4)/2 + 3)/2 + 2)/2 + 1) = 5$$

The program is Listing 1 will handle this particular case and some others, but would need refinement for more general application.

The equation is input as a string A$. The string is scanned by subroutine 1000, starting at the right hand end applying the following rules:
- If a numeric code is found, the subroutine sets the variable E to the number, treating it as decimal.
- If an '=' sign is found, E is copied into N.
- If a ' *' sign is found, N is divided by E.
- If a '+' sign is found, E is subtracted from N.
- If a '−' sign is found, E is added to N.
- If a '/' sign is found, N is multiplied by E.
- If an alphabetic character is found, it is set in R$.

When the scan reaches the start of the line, the contents of N are output as the value of R$.

This relatively simple routine suffices in the given case because the location of the brackets removes any doubt regarding priorities; each number and operator can be processed in the order in which it appears. For more general use, it would be necessary to take brackets and priorities into account. For example, the equation might be;

$$N+(4+6)*3 = 40$$

As it stands, the program would go sadly astray, giving the result as N = 3.33333. It fails to note the implication of the brackets, which mean that their their contents should be calculated first, the sequence being:

$$N+(4+6)*3 = 40$$
$$N+10*3 = 40$$
$$N+30 = 40$$
$$N = 10$$

Note that the multiplication must be processed before the addition, in accord with the usual order of priorities. However, with a slightly different equation, the process becomes:

$$N*(4+6)+3 = 43$$
$$N*10+3 = 43$$
$$N*10 = 40$$
$$N = 4$$

Since the multiplication is within the term containing the variable, it is processed after the addition. In a rather broad way, it can be said that each term of the equation needs to be resolved before applying the program given. This could be achieved by suitable analytical logic, but the processes involved would be quite complex, and there is (fortunately) no room to investigate that here. Nevertheless, the concepts involved give an interesting indication of the complexity of mathematical expressions.

ZOO—LOGICAL

Rather than pursue that line of thought too far, let us turn to pure logic problems, those which concern the association of sets of items. Given some of the associative links, you are asked to deduce the rest. In certain circles such problems are known as 'zebras', because a well-known example ends by asking 'Who owns the zebra?', but here is an original example concocted especially for the occasion.

'In each of four houses there are a husband and wife, a car and a dog. Charles lives at No. 1. Dave is married to Brenda. Ann, who does not live in the same house as the beagle, loves to drive the vintage Bentley which her husband bought last year. One dog is a collie. One man is Bob Bond. The Cooks own a boxer. Mr. and Mrs. Ames live at No. 3. Cathy lives at No. 2. Alan owns the beagle, Doris owns the corgi. Mr. and Mrs. Davy own a Ford. If the Austin is parked at No. 4, who owns the Talbot?'

This is not meant to be a particularly difficult example of this type of puzzle, but the real problem is to program your computer so that it can solve the problem for you. It will want a lot of help. Apart from learning the names of the various items and people, it will need to be taught which items are cars, which are dogs, and so on. Once it has all the data, however, it should be able to complete the job on its own.

The puzzle involves four sets, each of six items. These dimensions were chosen to suit the limitations of the Spectrum screen, but the addition of a fifth set makes such problems more interesting — and rather more difficult. If you make a good job of the program, the reason should be evident...

```
100 CLS
110 INPUT "Expression ? ",A$
120 LET D=0
130 LET E=0
140 LET S=0
150 LET N=0
160 DIM C(6)
170 LET O=LEN A$+1
180 GO SUB 1000
190 IF S=1 THEN GO TO 600
200 IF B>47 AND B<58 THEN GO TO 300
210 IF B=61 THEN LET N=E
220 IF B=42 THEN LET N=N/E
230 IF B=43 THEN LET N=N-E
240 IF B=45 THEN LET N=N+E
250 IF B=47 THEN LET N=N*E
260 IF B>64 THEN LET R$=CHR$ B
270 GO TO 180
300 LET D=D+1
310 LET C(D)=B-48
320 GO SUB 1000
330 IF S=1 THEN GO TO 600
340 IF B>47 AND B<58 THEN GO TO 300
350 LET E=0
360 FOR N=D TO 1 STEP -1
370 LET E=E*10+C(N)
380 NEXT N
390 LET D=0
400 GO TO 210
600 IF R$<>"" THEN GO TO 630
610 PRINT "No variable"
620 STOP
630 PRINT R$," = ";N
640 STOP
1000 LET O=O-1
1010 IF O=0 THEN LET S=1:RETURN
1020 LET B=CODE A$(O)
1030 IF B=32 THEN GO TO 1000
1040 RETURN
```

Listing 1. Program to back-solve expressions like the one in the first paragraph.

Stephen Renals

FLOATING POINT DICTIONARY

This dictionary for the Jupiter Ace will let you calculate logarithms to the bases e and 10, exponentials, powers, and one root of an equation using the Newton-Raphson method.

The Jupiter Ace has only the most basic of floating point words defined in ROM. However, with a little effort, a fairly acceptable package of floating point words may be defined. Indeed, this process is started by the inclusion of floating point stack manipulation words and square root and trigonometric functions in the Jupiter Ace manual.

The words 2 * and 2/ are machine coded words that perform the operations suggested by their names, included because they increase speed of operation. 2ROLL, 2PICK, 2?DUP, 2VARIABLE and 2CONSTANT are floating point variations of words that operate on ordinary 16-bit numbers (they can be used on double length integers, also). FLOAT has been defined so that signed 16-bit numbers may be transferred to floating point:
− 32 UFLOAT would leave 65504. on top of the stack
− 32 FLOAT will leave − 32. on top of the stack
FO< is the same as DO< given

in the ACE manual, since it also works on floating point numbers. FINPUT will wait for a floating point number to be entered, whereas INPUT (only included for completeness) will wait for a 16-bit number to be entered.

FI↑ will calculate powers of floating point numbers, so long as the index is an integer (in floating point form) not less than 0. It is similar to a 16-bit definition given in the ACE manual.

LN calculates the natural logarithm (log to the base e) of a floating point number. It does this by using the series:

$$\ln m = 2(h + \frac{h^3}{3} + \frac{h^5}{5} + \frac{h^7}{7} + \frac{h^9}{9} + \ldots)$$

where

$$h = \frac{m-1}{m+1}$$

This series gives a good approximation of ln m, when m is less than 10.

LN first splits the floating point number into mantissa and exponent. For example, 984.51 (= 9.8451E2) has a mantissa of

9.8451 and an exponent of 2. This is useful, since:

```
ln 9.84E2 = 2*ln 10 + ln 9.84
```

Hence the exponent is multiplied by ln 10 (2.30528) to be added on to the sum at the end. The value of h is then calculated from the mantissa and the above series is summed. The natural logarithm of a number may therefore be found.

LOG finds the log to the base 10 of a number, using the relation:

$$\log_{10} x = \frac{\ln x}{\ln 10}$$

EXP calculates the exponential function (exp x = e^x) of a floating point number. This is achieved by summing the Taylor Approximation to exp x:

$$\exp h = 1 + h + \frac{h^2}{2!} + \frac{h^3}{3!} + \frac{h^4}{4!} + \frac{h^5}{5!} + \ldots$$

where

```
n! = n * (n − 1) * (n − 2) * . . . * 1
```

LN is accurate to around 3 significant figures (SF). The size of the operand is not important, as LN splits the number, as mentioned earlier.

EXP is accurate to 5 or 6 SF on numbers between about − 2 and 12. It is accurate to 3 or 4 SF when the operand is between about 12 and 16, or − 2 and − 4. Above 22 of 23, or below − 6 it is not very accurate.

From these definitions, F↑ may be defined, which finds the power of any number. This word is based upon the relation:

$$a^b = \exp(b * \ln a)$$

F↑ is accurate to between 4 and 6 SF on answers ranging from 0.05 to 1.25 million, and accurate to 2 or 3 SF on answers up to 500 million or between 0.05 and 0.007.

Finally SOLVE will find the root of an equation using 26 iterations of the Newton-Raphson method. In this word, the function to be equated to 0 is contained in the word FX. F'X contains the derived function of FX. The derived functions of some common functions are given in Table 1.

Hence to use SOLVE to solve the following equation, for example:

```
9x⁵ − 4x⁴ + 2x = 3, ie
9x⁵ − 4x⁴ + 2x − 3 = 0
```

type in:

```
: FX (9x⁵ − 4x⁴ + 2x − 3)
    2DUP 5. FI^ 9. F*
    2OVER 4. FI^ 4. F* F−
    2SWAP 2DUP F+ F+ 3. F−
;
REDEFINE X

: F'X (45x⁴ − 16x³ + 2)
    2DUP 4. FI^ 45. F*
    2SWAP 3. FI^ 16. F*
    F− 2. F+
;
REDEFINE F'X
```

Typing SOLVE will leave the solution to $9x^5 - 4x^4 + 2x - 3 = 0$ on top of the stack (in fact SOLVE will leave .807163 on the stack).

These words occupy around 1100 bytes, with FX and F'X left 'empty', but including SIN, COS, TAN, and SQRT: hence the Jupiter Ace will require memory expansion to run it all. The words may be cleared by typing FORGET IT-PAL.

FX	F'X
$ax^m + bx^n$	$max^{m-1} + nbx^{n-1}$
$sin(ax + b)$	$a*sin(ax + b)$
$cos(ax + b)$	$-a*cos(ax + b)$
$tan(ax + b)$	$a/cos^2(ax + b)$
a^x	$a^x * \ln a$
$exp\ x$	$exp\ x$
$\ln x$	$1/x$

Table 1. Some common functions and their derivatives

Listing 1. The floating point dictionary for the Ace.

```
: IT-PAL ;

DEFINER CODE                 (Page 147 Ace manual)
DOES>
  CALL
;

: HEX 16 BASE C! ;

HEX

CODE 2*                      (N--N*2)
  DF C,                      (RST 24)
  CB C, 23 C,                (SLA E )
  CB C, 12 C,                (RL  D )
  D7 C,                      (RST 16)
  FD C, E9 C,                (JP(IY))

CODE 2/                      (N--N/2)
  DF C,                      (RST 24)
  CB C, 2A C,                (SRA D )
  CB C, 1B C,                (RR  E )
  D7 C,                      (RST 16)
  FD C, E9 C,                (JP(IY))

DECIMAL

Enter floating point stack rearrangement words - page 91 Ace
manual

: 2ROLL                      (N--)
  DUP 2* ROLL
  SWAP 2* ROLL SWAP
;

: 2PICK                      (N1--FP2)
  DUP 2* PICK SWAP
  2* 1+ PICK SWAP
;

: 2?DUP                      (FP--FP,FP if FP=0, FP--FP if FP=0)
  2DUP OR
  IF
    2DUP
  THEN
;

DEFINER 2VARIABLE            (4 byte variable)
  SWAP , ,
DOES>
;

DEFINER 2CONSTANT            (4 byte constant)
  SWAP , ,
DOES>
  2@
;

: FLOAT                      (N--FP signed UFLOAT)
  DUP 0< SWAP ABS
  UFLOAT ROT
  IF
    FNEGATE
  THEN
;

: F0<                        (FP--FLAG page 114 Ace manual)
  SWAP DROP 0<
;

: FINPUT                     (--FP)
  BEGIN
    RETYPE NUMBER
    4102 =
  WHILE
    DROP
  REPEAT
;
```

```
: INPUT                      (--N)
  BEGIN
    RETYPE NUMBER
    4181 =
  WHILE
    2DROP
  REPEAT
;

: FERROR?                    (FLAG--)
  IF
    8 15421 C! ABORT         (ERRNO = 8)
  THEN
;

: FI^                        (FP1,FP2--FP1^FP2; FP2=integer (not
                             less than zero)
  2DUP INT UFLOAT 2OVER F-
  OR FERROR?
  1. 2SWAP 2DUP OR
  IF
    INT 0
    DO
      2OVER F*
    LOOP
  THEN
  2SWAP 2DROP
;

: LN                         (FP--ln(FP))
  ?DUP 0< FERROR?            (can't have ln of a negative
                             number)
  DUP 256 / UFLOAT 65. F-    (extract the exponent)
  2.30528 F* 2SWAP           (a = exponent*ln10, m = mantissa)
  255 AND 16640 +
  2DUP 1. F- 2SWAP 1. F+ F/  (a, h = m-1/m+1)
  2DUP 2DUP F* 2OVER
  31 3                       (a, sum, h^2, h^{i-2})
  DO
    2OVER F* 2ROT 2OVER      (a, h^2, h^i, sum, h^i)
    I UFLOAT F/ F+           (a, h^2, h^i, new sum)
    2ROT 2ROT 2
  +LOOP
  2DROP 2DROP 2. F* F+
;

: LOG                        (FP--log_10 FP)
  LN 2.30258 F/
;

: EXP                        (FP--exp(FP))
  1. 2SWAP 1.                (1., h, 1.)
  31 1
  DO                         (sum, h, last term)
    2OVER I UFLOAT F/ F*     (sum, h, this term = last term*h/i)
    2ROT 2OVER F+ 2ROT 2ROT
  LOOP
  2DROP 2DROP
;

: F^                         (FP1,FP2--FP1^FP2)
  2SWAP LN F* EXP
;

Enter SIN COS TAN SQRT from page 92 of Ace manual

: FX                         (function equated to 0 by SOLVE)
                             (redefine to desired function)
                             (before executing SOLVE)
;

: F'X                        (derived function of FX for SOLVE)
                             (redefine before executing SOLVE)
;

: SOLVE                      (--FP Equates function FX to 0)
  1. 26 0                    (F'X is derived function of FX)
  DO
    2DUP FX 2OVER F'X F/ F-
  LOOP
;
```

Peter Green

PROCopinion

Two events involving the Electron took place last month, and have earned Acorn both brickbats and bouquets from the editor.

From little Acorns mighty oaks do grow, and Acorn have certainly come a long way since the early days when the company was a twinkle in the eyes of a few of the then Uncle Clive's employees. Now a major force in the British computer industry, and obviously destined to be one of the survivors in the cut-throat battles that now rage over the destination of your pocket money, they have unfortunately not advanced as far in the field of public relations as they have in electronics. At least they haven't judging by the press launch in August of the Acorn Electron

One of the mixed blessings in the life of a journalist is the Press Launch. We tend to get inundated with invitations to various binges to announce the arrival of new computers, peripherals, software, companies, marketing strategies, and sometimes apparently just to remind us that a business still exists. Often the PR company organising the reception has sent out invitations to all and sundry, and the product has no relevance to the target readership of the magazine, so the lure of free food and drink has to be resisted and the invite turned down. Sometimes you guess wrong, and come away feeling that you've wasted a couple of hours on something of no use to the magazine.

But when a PR company (and wild horses wouldn't drag the name of the Quentin Bell Organisation from my lips) sends out beautifully embossed invitations to meet Acorn Computers 'At Home' to see the new Electron, a major new computer from the people who brought you the BBC Micro, surely this will be time well spent?

How wrong can you be.

GONE TO LAUNCH

Since we're giving the Electron our usual extensive review in the January issue of *Computing Today*, let's, for a change, review the launch rather than the computer. The bigger the company and the product, the larger the hotel that is hired, and Acorn did things in style by hiring the Ballroom at the Park Lane Hotel, Piccadilly. An unusual touch as we went in was in the ritual issuing of identity badges (useful after over-indulgence in the wine). Normally they are cards with typed or written labels, but here guests typed their name and company into a computer which then produced a badge on a printer. (The fact that BBC Micros and not Electrons were used, rather gave away the fact that the Electron has no printer port).

After milling around for a bit to allow late-comers to arrive, we were asked to go through into the main hall. This was when the phrase 'At Home' started to make sense – one whole wall of the room had a life-sized facade of a house, Ideal Home Exhibition style, built in front of it. We took our seats, the lights dimmed, and out through the front door came Cliff Michelmore. He made his introduction, a slide-tape demonstration started running (projected on a screen on the 'garage door), and . . .

A loud female voice at the back of the hall interrupted, shouting that she didn't understand, the launch was patronising her and anyway she couldn't make sense of the jargon. As she walked forward we saw it was Wendy Craig; ho, hum, a put-up job. Mr. Michelmore, whose rate of delivery implied his autocue was running a bit slow, invited her up to the stage for a 'user-friendly' explanation of computers in general and the Electron in particular.

We were then treated to 40 minutes of rehearsed repartee in

Here we see the Acorn House, with Wendy Craig as the Devil's advocate and Cliff Michelmore scratching his head. A lot of journalists were scratching their heads too . . .

The Electron is a nice piece of hardware — and it can choose cocktails, too...

which we learned the history of Acorn (but what does the Electron offer?), we had a simulated dialogue between Wendy Craig and herself on the screen (but what does it *offer*?); we discovered that the computer can give cocktail recipes if you tell it what drinks you have (so can any other computer given the right program; *what does it offer*...?) In short, we learned a great deal more about Quentin Bell's estimate of the intelligence of computer journalists than we did about the Electron. At the end Mr. Michelmore explained that he had just given a user-friendly lecture for the computer novice, and that most of us there didn't fall into that category. So why waste our time with it? I felt patronised and insulted and so did many other people I spoke to afterwards.

Back outside, as we sipped our 'Electron cocktails' and watched Chris Curry of Acorn fending off feminist reporters asking if the Electron image was really one of dizzy blondes, there was at least the chance to get the information from the press pack: assuming you were lucky enough to get one. The Electron is better at cocktails than counting invitations because they ran out of press releases. An additional touch of madness was the inclusion in each pack of a piece of green Electron rock, specially made with the words down the centre. The cost of the whole launch must have been immense, and I can't help feeling it would have been cheaper, and better for the Electron, if Acorn had simply given everyone a computer at the door and said "Go away and play with it". Then Quentin Bell could get on with some useful work: a colleague rang them recently to ask for as many photographs of the BBC Micro as they had. They sent **one**, because they'd "run out".

The whole occasion was summed up for me as I was leaving, and stopped to alleviate the effects of the Electron cocktail. Colin Barker of *Which Computer* had peeled off his Acorn Electron Launch badge and stuck it on the wall of the lavatory. So say all of us.

ACORN USER EXHIBITION

Happily I can say much nicer things about the Acorn User Exhibition a few days following the press launch. This was anything but a waste of time: I went intending to spend a couple of hours and ended up staying all day. Spread over two floors of the Cunard Hotel in Hammersmith, this was an occasion to delight the heart of any Acorn user and leave the weak-willed with large holes in their wallets. The BBC Micro figured largely, of course, but there was already a trickle of products for the Electron which will have swelled to a flood by the time sales of the machine take off in earnest.

In fact the exhibition was almost too good, because it was packed with people and could really have done with wider gangways between stands. I visited on the Friday and I shudder to think what it was like at the weekend.

First port of the call was the Acorn stand, to investigate the whereabouts of the review Electron

And you thought I was joking. A stick of Acorn Electron rock was provided for each journalist — in lieu of decent information.

promised four weeks previously. It turned out that they'd contacted 18 or so magazines but only had eight computers spare, so don't worry — it happens to us as well as members of the public! Hopefully Acorn have stockpiled sufficient machines to prevent the all-too-familiar supply shortages that plague new machines; I left the poor man explaining to people where their add-on processor cards had got to.

The Acorn stand was quite impressive from a distance; there was a 'frieze' of monitors around the top with graphics and messages apparently scrolling sideways round and round the stand from one set to another. Very clever.

Software, particularly games software, for a machine with built-in sound effects tends to all sound the same. How many games for the Beeb feature the whoop-whoop-whoop noise that announces an extra life on Acornsoft's Defender? So it was nice to hear something a bit different at the show, emanating, I think, from the vicinity of the Broadway Electronics stand. With great ingenuity someone had made the Beeb sound just like a honky-tonk piano, and the demo was rattling off an intricate piece of boogie. Nice.

Microwriter's stand featured their 'cut-down' version which interfaces with the BBC Micro. The original idea of the Microwriter was for a handheld text editor with a five-finger keyboard for rapid entry — faster than handwriting, claim Microwriter. By removing everything except the keyboard circuitry they've produced a much lighter, and much cheaper, version that allows rapid text entry on a computer.

My only previous experience of the Microwriter was on (yet another!) press booze-up last year on a boat down the Thames, and although I tried hard I couldn't get the hang of the unusual keypad. (It took me years to figure out QWERTY). But at the exhibition a charming young lady took me though the alphabet, explaining the fairly sensible mnemonics, and within five minutes I was writing short sentences, albeit slowly. I can still remember most of them, too; for P you Press down with all five fingers, for S you only use the Signet ring finger . . . Maybe I'll buy one sometime and become more time-efficient by writing articles on the train.

Next I paid a visit to the stand of Brainstorm, who are Acorn and Torch dealers. Incidentally Brainstorm are the company who supplied the Torch Z80 disc pack for review in the September

The Microwriter is a handheld word processor with only five keys. Now it can speed up text entry on the BBC.

Computing Today, and who we ungratefully forgot to credit. Sorry, chaps. Lots of goodies here, including a new Adventure game for the BBC Micro and Electron which they were kind enough to give me a review copy of (after a lot of inconvenient plug-swapping and tape-recorder adjusting). I haven't had a chance to try it out yet, but it's called The Eye of Zoltan and we'll be reviewing it as soon as possible.

Brainstorm also had something called the Incredible Hulk. Not another arcade game, I groaned — and fortunately it isn't. Hulk is an Expert System program, written in BASIC and capable of some nifty tricks. Richard Forsyth, the author, was on the stand making a few minor improvements and put the thing through its paces for me. The example database used actual figures supplied by the Coal Board

for the chemical content of various coal samples, and with the first program module you invent rules that use the composition to deduce the location of the mine. The program checks the results of each rule and recommends whether to keep it or throw it out. When you're happy with the set of rules, you load another program module which tests unknown data using the rules and deduces, in this case, the mine location of each sample. In the demonstration the actual origins of each sample were compared with the computer's guesses, which were found to be about 90% accurate: when it's perfected, the Coal Board are going to be one of the first customers for this program, which can handle all manner of problems.

Interested parties can find Brainstorm at 103A Seven Sisters Road, London N7 7QN (phone 01-263 6926).

M. C. Hart

TAPE APPEND ROUTINE

Appending tape program files can be awkward but here we show you how to take the pain out of it.

A routine to append subroutines or to incorporate parts of other programs into a resident program is one of the most useful routines in any Toolkit — indeed one wonders how one ever manages without it. There ARE occasions, however, when a tape-append can be very useful eg if your PET does not have a Toolkit-type ROM installed or, if like myself, you have decided to upgrade your Toolkit by installing a chip such as COMMAND-0, only to discover that the append function will now only append from disc and not from tape.

There are at least two popular ways of appending tape program files but they are both a little 'messy' in their own ways:

(i) The Butterfield MERGE routine in which the subroutine to be appended is converted first into an ASCII file and then read back as if they were being entered via the keyboard. As a MERGE routine this is fine but it does have the disadvantage that subroutines have to undergo a conversion process first.

(ii) A Machine code APPEND (such as detailed in Raeto West's **'Programming the PET/CBM'**) operates by loading machine code, then loading the program to be appended, then 'hoisting' it into high memory, then loading the 'main' program and finally 'appending' the first program by bringing it down from high memory to tag onto the tail of the main program. If one is working from discs, this is fine as access time is quick but in tape the process is very much slower and prone to error.

The routine to be developed here works in the way that you might expect it to work: ie with a program already held in memory, it is possible to append another program straight onto the end of it as though one were using the Toolkit-type append. The routine is compact (some 5-6 lines of BASIC) and can be entered in direct mode if necessary. To explain exactly how it works a little knowledge of tape headers is required.

HEADERS EXPLAINED

When programs are stored on tape, the program header has the structure shown in Fig. 1. Locations 635-636 in the first cassette buffer store the 'start' address for the load and this is usually 01 and 04 (= 1025 decimal ie start of BASIC program area). Locations 637-638 store the 'end' address for the load. It is possible to examine the header for any tape without loading the main program by typing OPEN 1 (or more fully, OPEN 1,1,0, ''PROGRAM NAME'') and then examining the bytes in the monitor (SYS 4 and then M 027A-029A).

This tape-append routine works by opening the file of the subroutine and recomputing new 'start' and 'addresses' for the load so that the subroutine will tag onto the 'tail' of the resident program shown in Listing 1.

PROGRAM NOTES

Line 3 is included so that the newly augmented program will skip the APPEND routine and go direct to line 10. It can be added later if you wish. If you include line 3 then activate the APPEND with RUN4.

Your subroutine should start at line 10, preferably with a REM. If you leave a line 10 REM in your append program and later append another line 10 the program will finish with two line 10 REMs. Deleting line 10 will delete the first of these leaving the (newly appended) line 10 in place.

Your append routine can be used as many times as you like to append further subroutines. The responsibility is yours, however, to ensure that the numbers are in ascending order before the process is started.

To append in direct mode is fairly easy but you must be exact! (a), (b), (c), (d) represent lines typed in direct mode assuming a program resident in memory:

a) CLR:A=1025:B=256:FOR J=1 TO 1E6:
L=PEEK(A)+B*PEEK(A+1):A=L:
IF L<>0 THEN E=L:NEXT
b) OPEN 1 (or OPEN 1,1,0,"Program
Name") ** Tape must be positioned
c) N=PEEK(637)+B*PEEK(638)+E-1025:
POKE 635,E AND 255:
POKE 636,E/B:POKE 637, N AND 255
d) POKE 638,N/B:POKE 42,PEEK(637):
POKE 43,PEEK(638):
SYS?????:SYS?????:CLR

The SYS calls are:

```
BASIC 1 SYS62403:SYS50227
BASIC 2 SYS62393:SYS50242
BASIC 4 SYS62456:SYS46262
```

Disc appends are documented in
'*Commodore Computing*' (July
1982) p.4 for BASIC2;
'*Commodore Computing*'
(October, 1982) p.4 for BASIC4.
    I am sure you will find this
routine more simple in use than
other Tape-Append routines,
should you ever wish to use it.

```
3 GOTO 10:REM ** NEW PROGRAM STARTS
  AT 10
4 A=1025:B=256:FOR J=1 TO 1E6:
  L=PEEK(A)+B*PEEK(A+1):A=L:IF L<>0
  THEN E=L:NEXT
5 INPUT "APPEND NAME~S3 SPC][3 CLJ";
  A$:B$=A$:IF ASC(A$)=160 THEN A$="":
  B$=""
6 OPEN 1,1,0,A$:PRINT"[CD][REV]
  APPENDING ";B$:
  N=PEEK(637)+B*PEEK(638)+E-1025:
  POKE 635,E AND 255:POKE 636,E/B
7 POKE 637,N AND 255:POKE 638,N/B:
  POKE 42,PEEK(637):
  POKE 43,PEEK(638):IF PEEK(144)=46
  THEN 9
8 SYS 62456:SYS 46262:CLR:END:REM **
  BASIC 4
9 SYS 62393:SYS 50242:CLR:END:REM **
  BASIC 2
10 REM ** VARIABLE LIST
11 REM ** A=LOW BYTE (BECOMES LINK)
12 REM ** B=HIGH BYTE MULTIPLIER
13 REM ** L=LINK ADDRESS
14 REM ** E='END' OF RESIDENT PROGRAM
   (BECOMES THE NEW START ADDRESS)
15 REM ** A$=NAME OF PROGRAM TO APPEND
16 REM ** P$=COPY OF A$
17 REM ** LINE 8 LOADS FOR BASIC 4
18 REM ** LINE 9 LOADS FOR BASIC 2
```

Listing 1. The tape append routine.

	027A	027B	027C	027D	027E	.....					
	01	01	04	10	04	4E	41	4D	45	20	20	20
	634	635	636	637	638	639	640	....			

IDENTIFICATION No. | START ADDRESS | END ADDRESS | NAME (16 CHARACTERS LONG OR NAME + SPACE = $20)
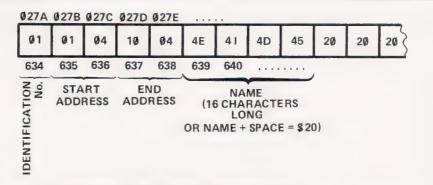
Fig. 1. The structure of the program header on tape.

LINE	FUNCTION
Line 4:	As BASIC programs end with three zero bytes, this routine works through the link pointers to ascertain the position of the middle zero byte. This is the 'start' position of the new program.
Line 5:	Requests the name of the subroutine (in A$), copies it into B$. Note the three shifted spaces and three cursor lefts in the input string so that a null return will result in A$ holding a **shifted** space (ASCII 160) and therefore not dropping into direct mode. If the tape to be appended is at a known position one can utilise this by pressing Return only.
Line 6:	OPENs the tape header of the named (or next) tape-file and prints out (in reverse) 'APPENDING (PROGRAM NAME)'
Line 7:	The low and high bytes of the new 'end' address are POKEd into position and the END-OF-BASIC pointers updated so that the composite program can be saved if necessary.
Lines 8-9	Enters a ROM routine to complete the load (the headers having been adjusted) and another routine to rechain the program lines (versions for BASIC4 and BASIC2). If your version is only going to be used on one version of BASIC then one of these lines can be dispensed with. The CLR and END statements *are* important! BASIC 1(Old ROMs) should substitute SYS 62403 : SYS 50227.

Table 1. How the program works.

Wendy J. Palmer

CLUB CALL

Read on to find user groups and clubs in your area.

PETERBOROUGH PERSONAL COMPUTER CLUB

920 Bourges Boulevard,
Peterborough,
PE1 2AN
Contact: Andrew Pike
Tel: 0733-44342 (after 5.0 pm)

Andrew Pike is the new Secretary of this club which meets fortnightly on Monday evenings at Crosfield Electronics Social Club. All machines and novices and experts are welcome.

LAMBETH COMPUTER CLUB

54 Brixton Road,
London SW9 6BS
Contact: Robert J. Baker
(Chairman)

Further to the mention we gave to the formation of this club in a previous issue of *Computing Today,* The Lambeth Computer Club has now been formally organised, even to the extent of a draft constitution. The inaugural meeting was held in April with some 12 members attending (60% of the membership). The first Annual General Meeting is planned for Thursday September 15. New members should contact Robert at the above address and your name and address will be passed as soon as possible to the Membership Secretary.

NAMEBUG

Contact: Dave Watts (Chairman) or Andy Purkiss (General Secretary)
Tel: 0245-358127 (Dave Watts) or 0376-515609 (Andy Purkiss)

This is a BBC Micro User Group for users in the North and Mid Essex area and meetings are held in a comprehensive school in Witham at 7.30 pm on the second Thursday of each month. The meetings follow a structured programme of talks and demonstrations by local dealers followed by a general forum. They also intend to start a workshop evening for members who would like help with modifications, upgrades and interfacing. The existing members come from as far afield as Chelmsford, Maldon,

Braintree, Colchester and Clacton so they would like to hear from any other user groups in those areas. Even if you don't own a micro but would like to see what they can do you are welcome to go along.

NATIONAL MZ-80K/A USER GROUP

48 Brander Road
Leeds
West Yorkshire LS9 6PR
Contact: Nigel Brown

Attention all Sharp MZ-80K/A owners nationwide. This newly formed group aims to produce a quarterly newsletter giving advice, tips, program listings and members' letters. Membership is free giving access to a members' library. When writing please state your machine type and all enquiries should be accompanied by an SAE.

SCOTTISH TRS-80 USERS' GROUP

72 Morningside Drive
Edinburgh EH10 5NU
Contact: Dick Mackie
Tel: 031-447 6651

Owing to a change of address any TRS-80 users in the Edinburgh area should note the new details for this group. The group meets at the Mansion House Hotel, Milton Road West, Edinburgh, on the second Thursday of the month and Genie and Dragon 32 owners are also welcome.

BRACKNELL COMMODORE CLUB

31 Pembroke
Hanworth
Bracknell
Berkshire
Contact: Paul Tilsley (Secretary)

July saw the first meeting of this club at The Admiral Cunningham Hotel, Priestwood Court Road, Bracknell. The club is offering to people of all ages discounts on hardware/software from local dealers, a swop shop, auctions for second hand items, competitions with cash prizes, teaching and advice for beginners or those with problems, a raffle each week (hopefully with hardware as

prizes), and a general meeting place where like-minded people can chat and where those thinking of buying a computer can find out more before they take the plunge. Membership is £5 per year (single) or £10 per year (family, regardless of number).

MANCHESTER ATARI COMPUTER ENTHUSIASTS

32 Edingburgh Road
Little Lever
Bolton
Lancashire
Contact: M. W. Davies
Tel: 0204-700757 or 061-799 0124
(Jack Young)

This recently formed group meet every other Thursday at 7.30 pm on the premises of The Ellesmere Hotel, Worsley Road, Worsley, Manchester

MID-CHESHIRE COMPUTER CLUB

75 Swanlow Lane
Winsford
Cheshire CW7 1JD
Contact: Simon Sadler
Tel: 060-65 53339

This group get together on the second Friday of every month at the main Winsford library in the town centre; meetings start at 7.30 pm and last until 10.00 pm. The meetings are varied in content and include various courses run by members for the benefit of newcomers. Once a year, usually in July they hold A 'Computer Olympics' which is purely a fun evening for members and the community. Newcomers can visit for two months without joining; membership is £1 per year plus 50p for adults and 20p for juniors (under 16) and pensioners.

WEST HERTS 80 USERS ASSOCIATION

24 Sempill Road
Hemel Hempstead
Hertfordshire HP3 9PF
Contact: R. C. Smith (Secretary)

The Association was initially formed to serve the needs of owners of the Genie and TRS-80 families of micros. It achieved a regular membership of about 100 and has been a very active group. It is now part of the BBC Computer Literacy Scheme. In order to maintain their membership and bulk purchase schemes and to serve the needs of the BBC users it has been decided to start a BBC unit sub-group. The aim is that they as an Association with a constitution, bank account, regular premises and organisational

experience will invite BBC owners who wish to have their own club, to get in contact with the Secretary. He will then arrange an inaugural meeting which he will chair. With interested parties present he will then arrange club nights, venue and times, etc. When the club is running they will then assist in election of officers and committee and advice on their rules and expenses. Once all this is done they will then give the newborn club the option (without their being present) whether they wish to go their own way or remain a sub-group for mutual accounting and other benefits. The Association's present venue is St Stephen's Parish Centre, Station Road, Bricketwood. So if you're interested in finding other BBC users or are involved with the Genie and TRS-80 machines, why not give the Association a go?

BRIGHTON, HOVE & DISTRICT COMPUTER CLUB
30 Leicester Villas
Hove
East Sussex BN3 5SQ
Contact: J. Smith (Secretary)

The club meets every second Wednesday at the Southwick community centre from 7.30 pm to 10.00 pm, and has members of all ages with a wide variety of machines and interests.

STOCKTON AMATEUR COMPUTER CLUB
60 Croft Road
Eaglescliffe
Stockton-on-Tees
Cleveland TS16 0DY
Contact: Peter Cheshire (Chairman/Secretary)
Tel: 0642-784819

Catering for a variety of machines, the club meets every Monday at the YMCA in Stockton from 7.00 pm to 9.00 pm. The meetings alternate weekly between programmer's evenings and workshop/games evenings. The minimum age is 11 years and adults are particularly welcome; there is a weekly subscription of 30p per meeting, with £4 per annum for adults (over 18), £6 for family and £2 for junior membership. Unemployed people and OAPs get half price membership.

SOUTHAMPTON AMATEUR COMPUTER CLUB
Crestwood Centre
Shakespeare Road
Boyatt Woods
Eastleigh
Hampshire

Contact: Paul Blitz
Tel: 042-15 69050

Computer users in the Southampton area should note the new club address; all other addresses are invalid. Also of importance is the correct address of the club's Computer Town UK! It is 181 Woodmill Lane, Bitterne Park, Southampton and the contact there is Roger Shears.

COSTA BLANCA COMPUTER CLUB
Montebello, 25
La Nucia-Alicante
Spain
Contact: Ed Kelly
Tel: 965-87 08 51

Meetings are held on each Saturday of the month. Time and place of the meetings is published in the Costa Blanca News, the local English language newspaper available each Friday. Club members range from beginners to advanced users. Meetings offer members the opportunity to discuss their interests and problems with the experts in the club. Demonstrations are given on different computers by owners or guest speakers. There is no charge for holiday-makers on the Costa Blanca desiring to attend meetings.

UK ATARI COMPUTER OWNERS CLUB
PO Box 3
Rayleigh
Essex
Contact: Ron

This club was formed in the very early days of the Atari computers in an attempt to bring the then very few and far between Atari owners together via a newsletter. Since then the club has prospered and now has over 1000 members throughout the country as well as overseas. The newsletters contain many informative articles and interesting features covering such subjects as player/missile graphics, display lists, hardware and software news and reviews plus selections of members' program listings. They try to cover all aspects of computing from basics to machine code programming. They also keep a library of members' programs for exchange. The membership fee is £3 per annum so if you're an Atari owner seems like you should give this club a try.

ZX CLUB
H-35D
Saket
New Delhi 110017

India
Contact: Arun K. Nath

It seems that Sinclair computers get everywhere and this club has been formed in New Delhi. Meetings are held on the second Saturday of every month.

LEICESTER APPLE USERS GROUP FOR HELP AND SUPPORT
7 Bude Drive
Glenfield
Leicester
Contact: Hazel Bown (Secretary)
Tel: 0533-875253

In case you hadn't noticed, the acronym for this group is LAUGHS, so they sound like a happy bunch! They meet at 8.00 pm on the first Wednesday of the month at the 'Airmen's Rest', Ratby Lane, Leicester. The annual subscription is £5 and their activities include informed lectures, dealer demonstrations and educational discussion with experts. Future meetings will include a Pascal evening, machine code, Modem, robotics and a Forum. So it looks as if this group will really give you something to get your teeth into — why not go along for a taste of their activities?

LEICESTER MICROCOMPUTING CLUB
Contact: Chris Clarke (Secretary)
Tel: 0533-673988

The club meets every second Monday in the month at 7.00 pm in the Medical Sciences Building, University Road, Leicester. The aim is to aid and encourage the free exchange of information and ideas on microcomputer hardware and software; they attempt to provide a 'main' feature at each meeting in the form of a formal lecture or demonstration of microcomputing equipment and software, other informal talks are given from time to time and there are always plenty of members' machines on display. The membership is £5 (adult), £3 (full-time student) and £7 (family). Your first visit if free and further visits as a non-member will be charged per session as £1 (adult), 50p (full-time student) and £1.40 (family).

If you would like to have your club mentioned on these pages just send me the details to:

**Club Call,
Computing Today,
145 Charing Cross Road,
London WC2H 0EE.**

COMPUTAMART

AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...

BERKSHIRE

Micro General — **PRINTER SPECIALISTS**
Call for advice on printer selection.
Epson & Microline stockists. Interfaces & cables available for all popular micros.
6 THE BIRCHWOODS, TILEHURST, READING.
TEL: 0734 25226

CHESHIRE

northern computers
Churchfield Rd., Frodsham. Tel: (0928) 35110
Open: 6 days 9-6. Retail and Wholesale. Apple II & III Atom, BBC, VIC20/64, Newbrain, Dragon 32, Electron, Spectrum. All accessories. Easy parking off M56.

Computer Junk Shop
We Buy, Sell, Break Computers & Peripherals
10 Waterloo Rd, Widnes, Halton. Tel. 051 420 4590

CLEVELAND

BRIERS COMPUTER SERVICES
Polytechnic Bookshop (at the Poly)
1 King Edward Square,
Middlesborough.
Tel: 0642 242017.
"Everything for the Genie computer".

CORNWALL/DEVON

A. B. & C. COMPUTERS (CT)
Duchy House, 6 Lower Aylmer Sq., St. Austell.
Tel: 0726 64463/67337
Wide range of popular Micros, Printers, books and accessories. We stock all U need — try us first for service and competitive prices

PLYMOUTH'S nascom DEALER
S & R BREWSTER LIMITED
86-88 Union St., Plymouth PL1 3HG.
Tel: 0752 665011   Open: 6 days.

DORSET

ZIPPY ELECTRONICS
West Dorsets Nascom Dealer
Business and Personal Computer Systems, Printers, Software.
Mail Order Service operated.
Access/Barclaycard accepted.

HAMPSHIRE

XITAN SYSTEMS   23 Cumberland Place, S'hampton
Tel: (0703) 38740. (C.R.A. member)
Open: Mon-Sat 9.30am-5.30pm (Sat 1pm)
OSBORNE in stock.

HERTFORDSHIRE

NEWBRAIN SPECIALISTS
HARDWARE & SOFTWARE
Printers, **EPSON, SHINWA, JUKI** etc. Monitors.
Tape Recorders CP/M Now Available
Mail Order Access.
also available "Newbrain Dissected" — book and tape
ANGELA ENTERPRISES
Tel: Stevenage (0438) 812439 anytime

NORTH KENT

MEDWAY COMPUTERS
141 NEW ROAD, CHATHAM.
Tel: 0634 826080
Open Mon-Sat 10 - 5. Closed Weds.
Most computers and software stocked.

HOME AND BUSINESS
COMPUTOR CENTRE
1 NEW STREET, DOVER.
0304 212433
Sinclair Specialist. Books, machines, software, add-ons. Plus software and books for BBC, Dragon, Atari, VIC.

LANCASHIRE

LANCASHIRE MICROS
51 QUEEN STREET, MORECAMBE.
Tel: 411435. Also open Sundays.
Sinclair, Commodore, Acorn/BBC + Dragon
Authorised dealer for Bug-Byte, Imagine, Quicksilva, Artic, Melbourne House, Silversoft etc

BUG-BYTE AUTHORISED DEALER
LEIGH COLOUR LABORATORY LTD
87 Chapel St., LEIGH,
Tel: 0942 607661
Open: 9 — 5.30.

GT. MANCHESTER

A ★★★★★★★★★★ F
★ A & F SOFTWARE ★
★★★★★★★★★★
830 HYDE ROAD, MANCHESTER.
Tel: 061 223 6206
Open: Mon-Fri 9.30-5.30, Sat 9.30-5.00
Full range of software/hardware for most micro computers. Credit cards accepted.

NSC COMPUTER SHOPS
29 Hanging Ditch, Manchester.
Tel: 061 832 2269
Open: Mon-Fri 9.30am-5.30pm.
Sat 10-5. Retail and Wholesale.

LINCOLNSHIRE

SHARP CENTRE
16 Melville Street,
Lincoln.
Tel: Lincoln 32379.
Open: 9am-5.30pm closed Wed.

LONDON

Henry's COMPUTER SHOP
404-406 Edgware Road, London W2 1ED.
Tel: 01-402 6822
Open: 6 days a week. Order by phone or call in and see for yourself.

BYTESHOP COMPUTERLAND
324 Euston Road, NW1.
Tel: 01-387 0505.
Open: Mon-Fri 9am-5.30pm.
(C.R.A. member)

SOUTH LONDON

CROYDON COMPUTER CENTRE
29a Brigstock Rd., Thornton Heath
Surrey. Tel: 01-689 1280
BBC, Acorn, NewBrain, Gen.e, Oric
Kaga Microvitek Zenith Monitors OKI
80, 82A + 84 Printers. Paper Ribbons.
Software etc. etc. BUY/HIRE

MERSEYSIDE

LYVERSOFT
66 LIME STREET, LIVERPOOL   Tel: 051 - 708 7100
Open: 6 days a week 9 till 5.30
We specialise in our own Vic 20 software. Also software for Dragon, Sinclair, BBC, Acorn, Texas etc. Plus peripherals

MIDDLESEX

L.B. ELECTRONICS
11 Hercies Rd, Hillingdon.
Tel: Uxbridge 55399 (24hr ans. service)
Open: 6 days, 9.30am-6pm, (lunch 1-2.15 except Sat)
Surplus equipment, memory, EPROMs etc. Also established mail order service.

NORFOLK

ANGLIA COMPUTER CENTRE
88 St Benedicts Street,
Norwich.
Tel: (0603) 29652/26002.
Open: 6 days 9am-5.30pm.

NORTHAMPTONSHIRE

NORTHAMPTON
HOME COMPUTER CENTRE
58A Wellingborough Road, Northampton.
Tel: (0604) 22539
Open: 6 days a week from 10 - 6

NORTHERN IRELAND

NEWBURN Ballycorry Co. Antrim
6 Days until 8pm. Gemini, Galaxy, Nascom, Acorn Computers. Discs, monitors, printers servicing, hire. Industrial Control, Accounts, Word Processors etc
WHITEHEAD 78330

COMPUTAMART

AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...

COMPUTING TODAY PRESENTS YOUR OWN 'WHERE TO BUY IT' GUIDE.

SCOTLAND

VICTOR MORRIS GLASGOW

TANDY TRS 80, VIC 20, VIDEO GENIE, APPLE PANASONIC, CUMANA, EPSOM ETC.

340 Argyle Street, Glasgow G2: 041 221 8958

COMPUTER CABIN
24 The Parade, Silverdale, Newcastle.
Tel: 0782 636911

VIC-20, ZX81 hardware and software. BBC, Atom, Spectrum software.

SUSSEX

CROWN
56-58 South Street,
Eastbourne.
Tel: Eastbourne (0323) 639983/20496
Open: 6 days 9am-5.15pm.

GAMER
24 Gloucester Road, Brighton
Tel: 0273-698424.
Open: Mon-Fri 10am-5.30pm,
Sat 9am-5.30pm.

TYNE AND WEAR

HCCS ASSOCIATES
533 Durham Rd., Low Fell,
Gateshead. Tel. Newcastle 821924.
Open: 6 days 9am-5.30pm (Sat
10am-5.30pm). Specialists in: Acorn,
BBC, Video Genie, VIC 20.

WALES

SIR Computers Ltd.
91 Whitchurch Road, Cardiff.
Tel: 0222 - 21341
BBC

STEVE'S COMPUTER COMPANY LTD
CASTLE ARCADE, CARDIFF
Tel: 0222 — 371578
OPEN: Mon — Sat. 9 — 5.30
FOR COMPONENTS TO COMPUTERS

BYTE·RITE
microcomputers
Llandaff R & TV Ltd.
24 - 26 High St, Llandaff, Cardiff.
0222 - 563760
NASCOM GEMINI LUCAS LOGIC. C A D
NETWORKS DRAGON PERIPHERALS

BUSINESS & LEISURE MICROCOMPUTERS
16 The Square, Kenilworth.
Tel: Kenilworth 512127.
Open: Mon-Fri 9am-5pm. ½ day Thur
(lunch 1-2). Retail and Wholesale.

LOOKING FOR MICROCOMPUTER HARDWARE OR SOFTWARE? LOOK NO FURTHER THAN COMPUTAMART!

WARWICKSHIRE

OEM MicroCentre Rugby
OEM COMPUTER SYSTEMS
9-11 Regent Street, Rugby CV21 2PE,
Tel: (0788) 70522/3/4
THE FINEST COMPUTER SHOWROOMS IN THE MIDLANDS

CARVELLS OF RUGBY LTD
3/7 Bank Street, Rugby CV21 2QE
The Acorn/BBC Specialists
We also stock Texas T199, VIC 20, Dragon 32
Commodore 64, Sinclair ZX81/Spectrum, Oric, Atari,
Disk drives, Epson, Juki, Shinwa & Olivetti Printers
and Books
Tel: Rugby 65275/6

YORKSHIRE

BRADFORD'S COMPUTER SHOP
at Thomas Wright (Bradford)
Ltd., Thorite House, Laisterdyke.
Tel: Bradford 668890.
Open: Mon-Fri 8.45-5.30. (Sat 12am)

SYSTEM-SOFT
THE BEST FROM THE REST
Freepost, Ackworth, Pontefract WF7 7BR
We stock an excellent range of software from leading
manufacturers for Spectrum, Dragon, Vic 20, BBC.
SAE for FREE catalogue   Mail Order Only

Please include my business details in the next available issue of Computing Today:

Business Name: ....

Address: ....

....

....

Tel. No.: ....

Open Hrs: ....

Contact (Office Use Only): ....

Post To: Computamart, Computing Today, 145 Charing Cross Rd., London WC2H 0EE.

ONLY £17.50!

SHARP

WIN THE 48k POOLS?

S P E C T A D R A W 2

SPECTADRAW 2 is a Pools Prediction Program for the 48K ZX Spectrum Microcomputer.

The program looks at the recent form of the teams playing in each week's English and Scottish football league matches and then refers to a large database to see what has happened in the past when teams with similar form met. By combining the lessons learnt from the past with the teams' current league position, the program generates a draw probability factor for each match. It can then identify the matches which are likely to yield draws and output suitable predictions. It will also output the least likely draws for the benefit of those who prefer the fixed odds lines on coupons.

The program is supplied complete with a database tape containing data on over 7500 matches and a comprehensive instruction manual.

Program, database (both on high quality cassette) and instruction manual **£12.95** inclusive. (Cheques/PO's payable to B.S. McAlley).

SPECTADRAW
1 Cowleaze, Chinnor, Oxford OX9 4TD

AD INDEX

THE VALLEY

© ASP LTD 1982

What are you . . . Barbarian or Wizard?

Choose your character type carefully . . . Barbarians recover quickly but their magic doesn't come easily. A Wizard? Slow on the draw and slow to mature . . . but live long enough and grow wise enough and your lightning bolts are almost unstoppable . . .

DRAGON AND BBC A VERSIONS NOW AVAILABLE!

The Valley is a real-time game of adventure and survival. You may choose one of five character types to be your personal 'extension of self' to battle and pit your wits against a number of monsters. Find treasure, fight a Thunder-Lizard in the arid deserts of the Valley, conquer a Kraken in the lakes surrounding the dread Temples of Y'Nagioth or cauterise a Wraith in the Black Tower. In fact, live out the fantasies you've only dared dream about. **BUT BEWARE. . .** more die than live to tell the tale.

You've read the program (Computing Today — April '82 . . . Now buy the tape. Tape versions (£11.45 each inc P&P and VAT) available for: ZX Spectrum (48K), Atari 400 and 800 (32K) Dragon, BBC Model A and B, Sharp MZ-80A, VIC-20 (with 16K RAM pack). Disc version (£13.95 each inc P&P and VAT) available for: Apple II (DOS 3.3), Sharp MZ-80A and PET 8032 (8050 drives). Full instructions are included with the game, but if you want more detail on the program, a 16 page reprint of the original 'Computing Today' article is available at £1.95 all inclusive.

Fill in the coupon and return it to:

ASP Software, ASP Ltd, 145 Charing Cross Road, London WC2H 0EE

and become one of the many to play. . . The Valley. . .

Please send me the following versions of
The Valley Tape . . . . @ £11.45 all inclusive of P&P and VAT.
Disc . . . . @ £13.95 all inclusive of P&P and VAT.

I enclose my cheque/Postal Order/International Money Order (delete as necessary) for: £ . . . . . (Made payable to ASP Ltd) or Debit my Access/Barclaycard (delete as necessary)

Please use BLOCK CAPITALS

NAME(Mr/Mrs/Miss) . . . .

ADDRESS . . . .

. . . . POSTCODE . . . .

Signature . . . . Date . . . .
CT Nov '83
Please allow 21 days for delivery

TRADE ENQUIRIES WELCOME